E U R O P E A N S O U T H E R N O B S E R V A T O R Y

Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral

Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

VERY LARGE TELESCOPE

Data Flow System

High Level User's guide

Doc.No. VLT-SPE-ESO-19000-1780

Issue 3

Date 20/3/02

49 pages

Prepared	SEG	20/3/02	
	Name	Date	Signature
Approved	M.Peron		
	Name	Date	Signature
Released	P.Quinn		
	Name	Date	Signature

CHANGE RECORD

Issue	Date	Affected Paragraph(s)	Reason/Initiation/Remarks
1	02402/99	All	Preliminary draft
2	3/20/02	All	Complete version to be consistent with dfs-4_5
3	20/3/02	All	internal comments from the DFS Group

LIST OF TABLES

LIST OF FIGURES

the logical view of DFS at Paranal 13
Process distribution 18
OHS interfaces 20
Interfaces of the Olas Ws (wuXdhs) 22
Interfaces of the User workstations (wgsoffX) 24
Interfaces for the pipeline workstations wuXpl 25
Interfaces for the asto ws (wgsarc) 26
The standard renaming scheme 27
dataSubscriber login window, vs 01 35
dataSubscriber, main window vs02 36
DataSubscriber.PreferenceWindow 37
Gasgano, Main window 40
dfslog, main window with database error 47

TABLE OF CONTENTS

1 Introduction 7
1.1 Scope .7
1.2 Intended Audience .7
1.3 Applicable Documents .7
1.4 Abbreviations and Acronyms .8
1.5 Glossary .8

2 DFS overall concepts 11

3 DFS-UTx Architecture 13
3.1 Logical view of the DFS .13
3.1.1 Observation Handling .13
3.1.2 OLAS .14
3.1.3 Pipeline .15
3.1.4 Quality Control .15
3.1.5 ASTO .15
3.2 DFS main data structure .15
3.2.1 Observation Block .15
3.2.2 Raw frame .16
3.2.3 Reduction Block .16
3.2.4 Pipeline Products .16
3.2.5 Relations between the main data structures .17
3.3 Physical view of DFS .18
3.3.1 Olas & OHS workstation (wuXdhs) .19
3.3.2 The User workstations (wgoffX) .22
3.3.3 The Pipeline Workstation (wuXpl) .24
3.3.4 The Asto workstation (wgsarc) .25
3.4 Frame Renaming Scheme .26

4 Using the Observation Handling tools (wuXdhs) 29
4.1 Using P2PP .29
4.1.1 P2PP for visitor mode OBs .29
4.1.2 P2PP in engineering mode .30
4.2 OT for service mode OBs .31
4.3 Troubleshooting .32
4.3.1 Problems communicating with BOB .32
4.3.2 Problems communicating with the OB repository .34
4.3.3 BOB complains about a missing template .34

5 Using the User Workstations (wgsoffX) 35
5.1 Subscribing to Raw Frames and Pipeline Products 35
5.2 Re-subscribing to Raw Frames or Pipeline Products 38
5.3 Managing the currently delivered Raw Frames and/or Pipeline Products 39
5.4 Troubleshooting 40
5.4.1 Raw Frames and Pipeline Products are not delivered any longer 40
5.4.2 problems to start/stop the RAW and/or REDUCED data subscription 40
5.4.3 Some Pipeline Products seem to be missing 41
5.4.4 dataSubscriber complains not being able to connect to database 41
5.4.5 wrong subscription to RAW and REDUCED data 42

6 Using the Pipeline Workstations (wuXpl)) 43
6.1 Subscribing to Raw Frames 43
6.2 Starting the Data Organizer 43
6.3 Starting the Reduction Block Scheduler RBS 43
6.4 Creating Pipeline Products 44
6.5 Supplying Pipeline Products to the User Workstations 44
6.6 Troubleshooting 45
6.6.1 The Data Organizer is not receiving any frame 45
6.6.2 The Data Organizer dies during initialization 45
6.6.3 trouble shooting for the UVES pipeline 45

7 Using the dfs logging tool dfslog (wuXdhs) 47
7.1 Troubleshooting 48
7.1.1 dfslog fails to reconnect to the server 48
7.1.2 dfslog fails to read in tree graph: treegraph 48

1 Introduction

1.1 Scope

This document is a high level user's manual for the DFS as installed and configured at the VLT. It is to be consistent with the dfs-4_6 release or later.

This document does not contain any installation notes (installation procedures for DFS are described in [1]).

It is not intended to be a detailed user manual of every tool/process belonging to the DFS, but rather to give an overview of what is available where. References to the appropriate documents are available in the different chapters.

1.2 Intended Audience

This manual is intended for:

1. Astronomers carrying out observations in Visitor Mode
2. ESO staff carrying out observations in Service Mode
3. Paranal or La Silla personnel operating the DFS
4. any person wishing to get an overview of the Data Flow System and its main interfaces

It may be used by the archive operators on Paranal who need to understand the flow of data generated by the VLT. However, it does not describe any of the script/tools which have to be used to check database consistencies and to prepare user CDs.

The document does not describe the configuration of the DFS at ESO headquarters and therefore it is not intended for USG members preparing queues of Observation Blocks or for Quality Control staff running off-line pipeline tools and preparing quality control reports in Garching.

1.3 Applicable Documents

[1] VLT Data Flow System Installation Guide, VLT-SPE-ESO-19000-1781

[2] P2PP Users' Manual, VLT-MAN-ESO-19200-1644

[3] OT Users' Manual, VLT-MAN-ESO-19200-xxx

[4] OLAS Users' Guide, VLT-MAN-ESO-19400-1785

[5] OLAS Operator's Guide, VLT-MAN-ESO-19400-1557

[6] ASTO Operator's Guide, VLT-MAN-ESO-19400-1784

[7] Data Flow Pipeline and Quality Control Users' Manual, VLT-MAN-ESO-19500-1619

[8] FORS Pipeline and Quality Control Users' Manual, VLT-MAN-ESO-19500-1771

[9] ISAAC Pipeline and Quality Control Users' Manual, VLT-MAN-ESO-19500-1772

[10] UVES Pipeline and Quality Control Users' Manual, VLT-MAN-ESO-19500-2019

[11] GASGANO User's Manual, VLT-PRO-ESO-19000-1932

[12] DFSLog User's Manual, VLT-MAN-ESO-19000-1827

[13] Data Interface Control Document, GEN-SPE-ESO-19400-794

1.4 Abbreviations and Acronyms

The following abbreviations and acronyms are used in this document as listed below:

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
BOB	Broker for Observation Blocks
CLI	Command Line Interface
DO	Data Organizer
DFS	Data Flow System
DMD	Data Management and Operations Division
ESO	European Southern Observatory
FITS	Flexible Image Transport System
FTP	File Transfer Protocol
GUI	Graphical User Interface
LTS	Long Term Schedule
OB	Observation Block
OHS	Observation Handling System
OPC	Observing Programmes Committee
OT	Observing Tool
P2PP	Phase 2 Proposal Preparation Tool
PI	Principal Investigator
RB	Reduction Block
RBS	Reduction Block Scheduler
SM	Service Mode
STS	Short Term Scheduler
TCP/IP	Transmission Control Protocol/Internet Protocol
VCS	VLT Control Software
VLT	Very Large Telescope
VM	Visitor Mode

1.5 Glossary

Astronomical Site Monitor (ASM): hardware and software system which collects observing conditions such as: air temperature, air pressure, humidity, wind speed, wind direction, seeing, sky brightness/emissivity, sky transparency, precipitable water vapour content, and dust content of ambient air. Five-minutes averages of the measurements obtained are logged and periodically transfered to the OLAS system via the VLT Control software in order to be archived into the Ambient database.

Acquisition Template (AT): an *Observation Block* object. An AT is used to specify how a target will be acquired by the telescope. It may also specify any preliminary instrument configuration steps (e.g. set rotator to specific angle). It can contain parameters for interactive as well as automatic acquisitions. This template may define a different instrument/detector configuration from the templates within the Observation Description. Each science OB contains at most one AT.

Archive Storage System (ASTO): Archive Storage System providing means for storing data onto a long-term archive media (CDs or DVDs).

BOB (Broker of Observation Blocks): VCS tool which receives OBs from the OHS applications (OT or P2PP). BOB accepts the incoming OB on the VCS side and begins execution.

Calibration Database: Database containing master calibration data.

Calibration OB: OB used to acquire calibration data. Such OB does not contain any AT.

Constraint Set (CS): an *Observation Block* object. A CS lists observation conditions required for the OB execution (i.e. requirements for sky transparency, seeing, airmass, lunar illumination, and moon

angular distance). Each Observation Block contains at most one CS.

DO: Data Organizer, tool of Pipeline which classifies and analyses the content of any incoming raw frame and creates the corresponding Reduction Block (RB), if appropriate. Assembles calibration frames and raw data to be processed following data reduction recipes (data reduction procedures) specified in a RB.

Exposure: a synonym for the acquisition of a single data frame, typically resulting in a single FITS file.

Instrument Package (IP): set of files containing the TSF and ISF files for a specific instrument. The correct IP must be installed before OBs can be created for that instrument.

Instrument Summary File (ISF): part of the IP, contains a summary of the P2PP addressable optical elements of that instrument.

Master calibration product: a reduced frame used as a master for calibration of science and calibration raw frames.

OB Repository: Database containing two kinds of Observation Blocks: (1) Service Mode Observations Blocks which are first submitted to ESO for review and scheduled via the OT tool, then for possible execution; (2) Visitor Mode Observation Blocks which are stored only when being submitted to the VLT Control Software for execution.

Observation: a coordinated sequence of telescope, instrument, and detector actions that results in a scientific or technical dataset.

Observation Block: Smallest observational unit within the Data Flow System. It contains a sequence of high-level operations, called 'templates' that need to be performed sequentially and without interruption in order to ensure the scientific usefulness of an observation. Observations Blocks may contain scheduling requirements. They are used both in *Visitor* and *Service* Mode to acquire data.

Observation Tool (OT): Tool used to create queues (sets) of *Observation Blocks* for later scheduling and possible execution.

Observing Run: an approved ESO programme consists of one or more Observing Runs, each of which specify an independent combination of telescope, instrument, and observing operations mode (i.e. *Service Mode* or *Visitor Mode*).

On-Line Archive System (OLAS): System responsible for receiving and distributing all data products generated by the VLT and by the on-line pipeline.

Phase 2 Proposal Preparation Tool (P2PP): Tool used to create and (in visitor mode) execute Observation Blocks.

Pipeline product: Result of the execution of a Reduction Block.

QC0: Quality Control level 0. On-Line tool that checks whether service mode OBs have been executed under the conditions specified by the astronomer. QC0 is executed on raw data.

QC1: Quality Control level 1. QC1 consists of quality checks on pipeline-processed data. The QC1 parameters are used to assess the quality of calibration products and the performance of the instrument.

Raw Frame: Result of OB execution by the VCS, i.e. immediate result of an exposure. Raw frames are delivered to the Science Archive and the Reduction Pipeline as FITS files. The headers (set of keywords) contain all info relevant for reduction, QC and archiving, in particular the identification of OB to which the exposure belongs. As they move through DFS, info is added to the headers (archiving information, seeing conditions, ...). They are stored in directories whose name has the format YYYY-MM-DD, where the date is that of the *night* to which the frame belongs, i.e. the noon preceding the exposure.

Reduction Block Scheduler (RBS): tool which schedules and executes RBs created and sent by the DO. RBS sends the RB to the DRS (MIDAS) which will actually perform the reduction.

Reduction pipeline: Subsystem of the DFS in charge of pipeline processing. Applies reduction recipes and its parameters (calibration frames) on raw frames to generate pipeline products.

Reduction recipe: standard procedure for reducing observational data in a standard way. Recipes are implemented for each of the instrument standard templates. Those scripts take as input raw frames and execute them in a particular Data Reduction System (DRS).

RTD: Acquired data is displayed via Quick-Look tools such as the Real-Time Display directly from the Instrument Control Systems.

Service Mode: observing operations mode where astronomer submits a detailed description of his/her observing programme to ESO for later possible execution. Service mode programmes are executed primarily in order of their OPC assigned priority but only when the astronomer specified observing conditions are achieved on-site.

Template: a high-level data acquisition operation. Templates provide means to group commonly used procedures into well-defined and standardized units. They can be used to specify a combination of detector, instrument, and telescope configurations and actions. Templates have input parameters described by a template signature, and produce results that can serve as input to other templates. As an example, an Acquisition Template takes target coordinates and produces through an interactive procedure the precise positions used later, e.g. to place an object on a slit.

Template Signature files (TSF): files which contain template input parameters used to create OBs.

VCS (VLT Control software): the software and hardware tools that are used to control directly VLT instruments, telescopes, and related hardware. It enables and performs the acquisition of scientific and technical data.

Visitor Mode: observing operations mode where the astronomer is present at the telescope when his/her observing programme is being executed.

2 DFS overall concepts

The Data Flow System (DFS) includes components for the preparation and scheduling of observations, archiving of data, pipeline data reduction and quality control.

The DFS supports the two observing modes handled by the VLT, namely *service mode* and *visitor mode*.

In service mode, observations are scheduled and executed by ESO staff with the objective of taking the best advantage of environmental conditions.

Visitor mode programs are allocated in advance a number of observing nights at the telescope: in that case, observations are scheduled and executed directly by visiting astronomers.

One of the VLT requirements is to provide its users with an easy-to-use system, e.g. to allow a simple specification of observations. Standard observing techniques, e.g. mosaics, jitter imaging are implemented as sequences of operations on the equipment and combined into so-called templates for which only few parameters have to be provided. These standard procedures will generate standard products with predictable properties.

The Observation Block (OB) is the atomic observational unit for the users of the VLT. OBs are created by the 'P2PP' tool.

An OB combines all the information necessary to define and execute a set of tightly related exposures which are needed to obtain a coherent set of data. It can be seen as the synthesis of the target information and telescope/instrument operations broken down to a set of templates.

Observation Blocks are then sent and executed by the VLT Control Software via the P2PP tool (visitor mode), or the OT tool (service mode).

As a result, one or more raw frames are created and delivered to the On-Line Archive, which in turn updates the 'Observation' database. Data are recorded by the ASTO sub-system on CDs or DVDs, and shipped to Garching (via diplobag).

The raw frames are processed in an automatic way by the reduction pipeline. Pre-defined calibrated solutions stored in the local Calibration Database are used to remove detector and instrument signatures. All information needed for processing is contained in the so-called Reduction Blocks (see 3.2.3. paragraph). As a result, one or more Pipeline Products are generated (see 3.2.4. paragraph).

Both raw frames and Pipeline Products are made available to the user on the User WorkStation.

3 DFS-UTx Architecture

This section addresses the architecture of the DFS as it is currently installed on any Paranal's UT.

3.1 Logical view of the DFS

The logical view shows the different subsystems of DFS at Paranal, without any mention of their implementation on existing hardware.

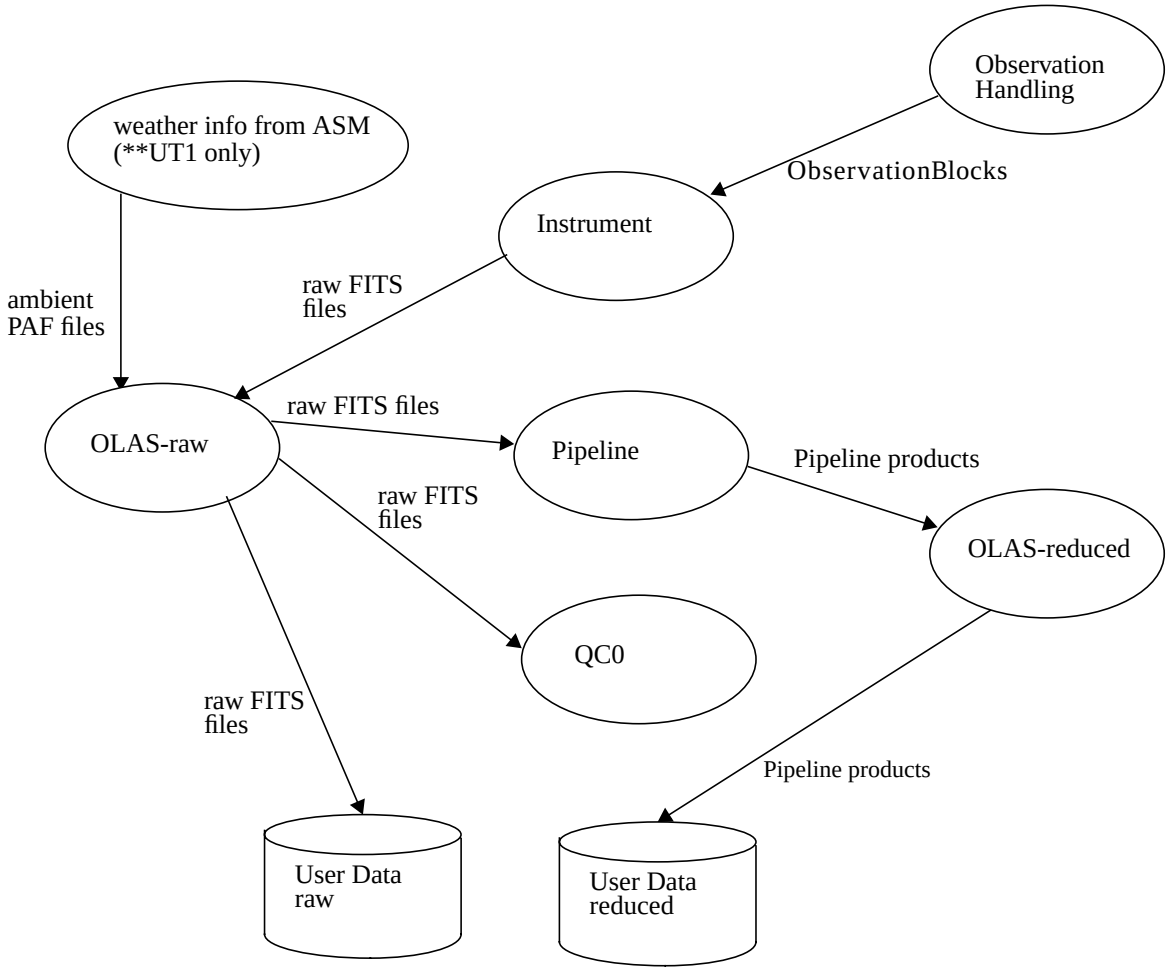


Figure 1: the logical view of DFS at Paranal

3.1.1 Observation Handling

The Observation Handling subsystem provides functionalities for:

- creating ObservationBlocks and submitting them for execution to the VLT Control Software through the **BOB** tool (which is running on the Instrument Workstation).
- creating and sorting queues of ObservationBlocks

3.1.2 OLAS

The On-Line Archive System (OLAS) takes care of receiving data, distributing it to interested parties and, if appropriate, storing the information into database tables.

An OLAS instance is a composition of at least one supplier (**vcsolac**) and one receiver (**dhs**), and of any number of subscribers (**frameIngest** and **dhsSubscribe**):

- **vcsolac** supplies the four types of supported data to dhs: FITS frames, PAF, LOG and OTHER files.
- **dhs** receives data from the supplier and distributes it to its subscribers
- **frameIngest** is one of the dhs subscribers. It is responsible for ingesting information into the existing databases. It takes care for instance of storing the information describing a FITS frame into the *Observation* database, and (for UT1 only) ingests the content of the PAF files generated by the Astronomical Site Monitor into the *Ambient* database.
- **dhsSubscribe** allows through a subscription mechanism the retrieval of any kind of data supported by OLAS.

Two instances of OLAS are running on each UT:

- OLAS-raw
- OLAS-reduced

3.1.2.1 OLAS-raw

The OLAS-raw subsystem processes data generated by the instrument, the telescope and (on DFS for UT1 only) the Astronomical Site Monitor (ASM) workstation.

- the Astronomical Site Monitor generates files in a **PAF format** and delivers them to OLAS-raw. Their content is extracted and ingested into different tables of the Ambient database. They are then delivered on request to ASTO.
- FITS files (i.e. raw frames) generated by the instruments are delivered to OLAS-raw. These files as soon as they have left the Instrument workstations get a unique name based on the value of the MJD-OBS keyword. Some of their FITS keywords are ingested into the *Observation* database to form the *Observation* database. The frames are delivered through a subscription mechanism to the Pipeline subsystem and to the user (on the **User workstation**). They are also delivered on request to ASTO.
- Operation Log files generated by telescope, ASM and instrument workstations are delivered to OLAS-raw, and then dispatched on request to ASTO.

OLAS-raw runs several instances of **vcsolac** and **dhsSubscribe**, one **dhs** and one **frameIngest**. The subsystem is depicted in Figure 2.

3.1.2.2 OLAS-reduced

The OLAS-reduced subsystem processes products generated by the Pipeline. These products are stored on disk as **FITS frames, binary tables or PAF files**. The topology of OLAS-reduced is very simple: it only runs one instance of **vcsolac**, one **dhs** and one **dhsSubscribe**.

Notice that header information of the pipeline products is not stored in any databases and therefore frameIngest is not involved in OLAS-reduced.

Furthermore, pipeline products are not archived onto long-term storage, and therefore they are not transferred to the ASTO workstation.

3.1.3 Pipeline

The pipeline subsystem is responsible for processing the raw frames generated by the instruments and for generating pipeline products. This subsystem consists of tools which are instrument generic (e.g. Data Organizer, Reduction Block Scheduler) and sets of pipeline recipes that are specific for each instrument. The local calibration database structure is instrument generic. For a given instrument, it contains configurations files and master calibration files.

The pipeline runs in an automatic mode in the on-line environment. It does not use the best calibration frames but the ones which are available in the local calibration database.

3.1.4 Quality Control

- Quality Control level 0 QC0:

The Quality Control level 0 checks are done on raw data and verify that the user-defined observational constraints have been respected during the observation. The data are checked for the following constraints:

  - Airmass
  - Moon Phase
  - Fractional Lunar Illumination
  - Seeing

- Quality Control level 1 QC1:

The QC1 parameters are measured by pipeline procedures on master calibration files and reduced files. QC1 parameters measured by the on-line pipeline are stored in the so-called QC1 Log Files.

3.1.5 ASTO

The Archive Storage System provides following functionality:

- store all data delivered by OLAS-raw onto long-term storage (CDs or DVDs)

3.2 DFS main data structure

3.2.1 Observation Block

An OB describes an observing sequence for one of the VLT instrument. It is composed of different parts:

- an Acquisition Template which specifies information related to target acquisition (e.g. rotator position angle, guide star). Such part is not required for calibration OBs.
- an Observation Description, which contains the instrument and telescope actions to be done within an OB. It consists of any number of *Template Signature Files* (TSF) which pass parameters to their corresponding *template script.*
- a Target Package, which contains the information unique to a specific target.
- a Constraint Set, which specifies the required observing conditions.
- Time Intervals, which describe optional absolute scheduling constraints.

Service Mode Observation Blocks are typically prepared several weeks in advance at home by astronomers (Principal Investigators). When OBs are ready for submission, they need to be stored

("checked-in") in a dedicated database, the OB repository, so that they can be reviewed by the User Support Group in Garching. They can then be scheduled for execution by ESO staff astronomers, thanks to the OT tool, when observing conditions are appropriate (as specified by the programme Principal Investigator). The content of this database located on a server in Garching is replicated in real-time into the OB repository located onto a database server at Paranal. If a user needs to revise an OB, it must first be "checked-out" (retrieved) from the OB repository.

Visitor Mode Observation Blocks may also be prepared at home by astronomers, but stored on disk as ASCII files (IMPEX files), and as binary files (via a local cache). They can not be stored into the OB repository, but instead corresponding files need to be transported and re-installed at the mountain by the visiting astronomer who will then re-import them, make some last modification and directly control their execution in order to cope with the vagaries of weather conditions. Anyway, those OBs also get stored into the OB repository at Paranal, but only after being submitted for execution to the VLT Control Software.

See 5.1 and 5.2 paragraphs of [2] for more information about the basic Observation Block concepts. Paragraph 5.2 of same document gives a very detailed description of an OB structure.

3.2.2 Raw frame

Raw Frames generated by VLT instruments are stored on disk as FITS files. Their header contain keywords which are compliant to the corresponding instrument dictionaries (see [13] for the official keyword definition). They contain all the information which is relevant for data reduction, quality control and archiving. Raw frames get an instrument-specific file name on the instrument workstation; as soon as they reach the OLAS system, raw frames are renamed and get a so-called archive unique name. This name is then kept across the whole system, while the original name is inserted in the frame header. As the files travel through the DFS, information is added to the headers. It includes archiving information (e.g. name of the file on the instrument workstation) and keywords describing environment conditions at the time of the observation (e.g. moon distance).

3.2.3 Reduction Block

A Reduction Block contains all the information needed for processing one or more frames. It includes:

- the name of the instrument
- the Reduction Recipe which has to be applied,
- the name and path of the output files (i.e. pipeline products)
- the name and path of the input frames
- the name and path of all master calibration files.

The Reduction Block is created by the DO software, which is responsible for the classification of the incoming frames, as an ASCII file.

3.2.4 Pipeline Products

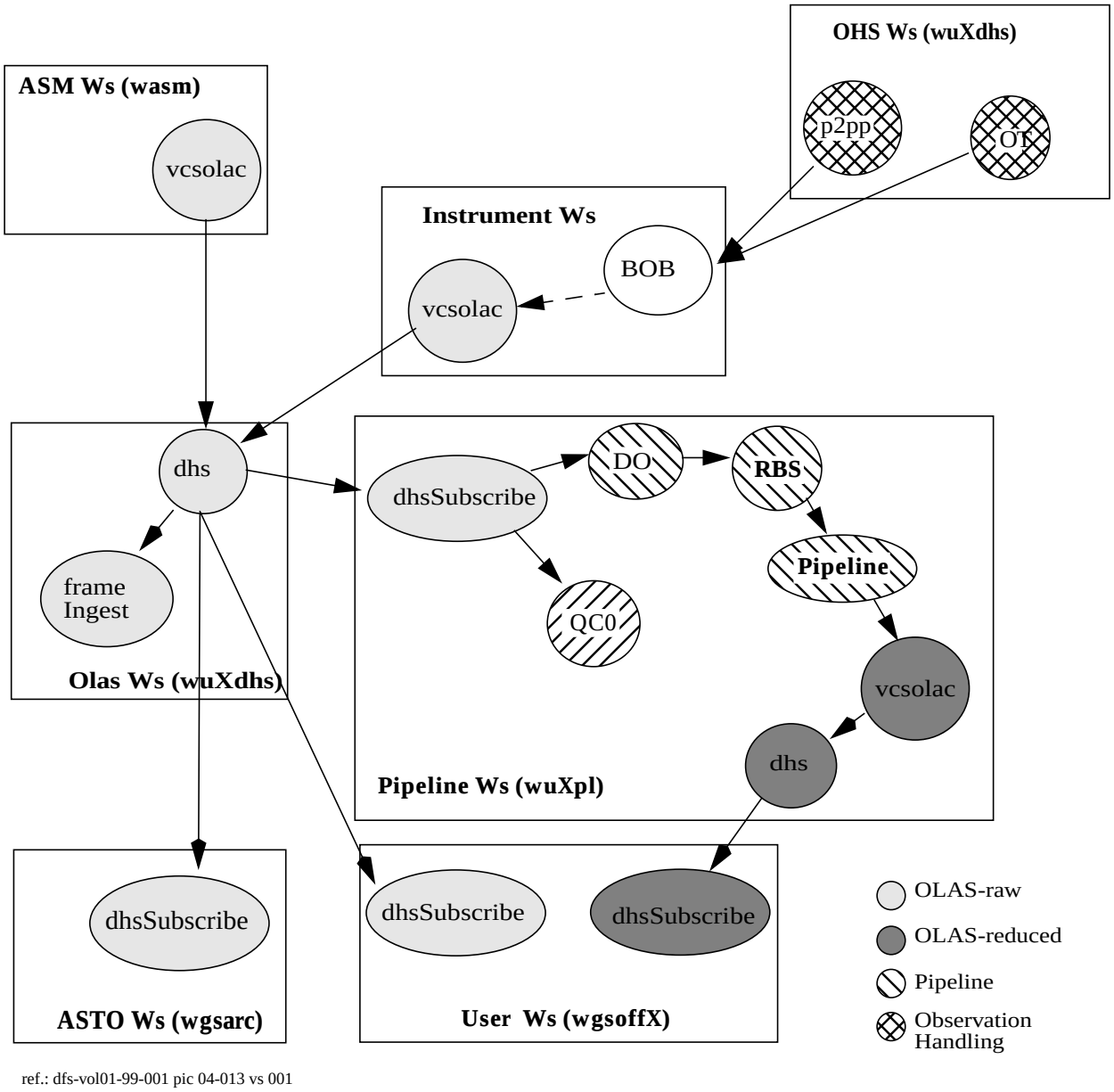
Pipeline Products are created as a result of the execution of Reduction Blocks. Those products are stored on disk as FITS files (image or table) or as PAF file. The headers of the FITS files contain all the information necessary to identify the reduction recipe, the input frames and the master calibration frames used for their creation.

3.2.5 Relations between the main data structures

- The execution of a template generates one or more raw frames
- The execution of an Observation Block therefore generates one or more raw frames
- A Reduction Block might be generated for each raw frame and for all the frames belonging to a template. This means that there is no one-to-one relation between an Observation Block and a Reduction Block.
- One or more pipeline products might be created as the result of the execution of a Reduction Block.

3.3 Physical view of DFS

The physical view shows how processes are organized on the existing workstations/servers.



ref.: dfs-vol01-99-001 pic 04-013 vs 001

Figure 2: Process distribution

Notice that the OHS and Olas Ws are currently the same machine.

Furthermore, thanks to the "UT-less" configuration of DFS, an User workstation can subscribe to any UT, i.e. can retrieve data acquired by any telescope. There are currently four User Ws, each being able to connect to one of the four Pipeline Ws and one of the four Olas Ws.

3.3.1 Olas & OHS workstation (wuXdhs)

3.3.1.1 Observation Handling Workstation (wuXdhs)

The Observation Handling workstation **wuXdhs** holds the following tools:

- P2PP
- OT

- **Tools & Processes**

P2PP is a GUI application which is used for creating, modifying, storing, and submitting for execution Observation Blocks to BOB. More precisely, main P2PP functionality is:

- retrieval of *Observing runs* definitions
- OB creation and viewing
- OB verification (calling the EVMs scripts, which are specific scripts delivered by the User Support Group in Garching; those scripts allow to perform further check about OB correctness)
- check-in OBs into the OB repository, and check-out. Once an OB is checked-in it can not be updated by P2PP; the user needs first to check it out to do so.
- OB repository browsing
- import and export OBs via IMPEX files. The IMPEX format allows exchange of OBs between different P2PP configurations.
- submit OBs to BOB for execution
- display and save OBs in OBD file format. The OBD format is used to manually send an OB to BOB for execution, if the BOB/P2PP connection (using the CCS environment) is not working properly.
- reports generation: ObsBlocks breakdown, Execution Time, OB Verification reports.

OBs are created using the instrument specific *TSFs* (Template Signature Files). A Template Signature File contains the list of input parameters to a *template* script, associated ordering and allowed ranges and default values. P2PP displays to the user the list of available TSF and allows him/her to select a TSF, enter parameter values, verify required ranges, and report possible errors. Only the names and the values of TSF parameters are stored: on disk, to the so-called Local Cache, and in the OB repository, if OBs are checked-in.

A general presentation of P2PP can be found at http://www.eso.org/~amchavan/papers/spie-2000.ps

The **Observing Tool (OT)** is used to create, modify and store observing queues, i.e. to select OBs from the OB repository, schedule them into ordered queues, so that they can be submitted for execution to BOB.

More precisely, OT allows to:

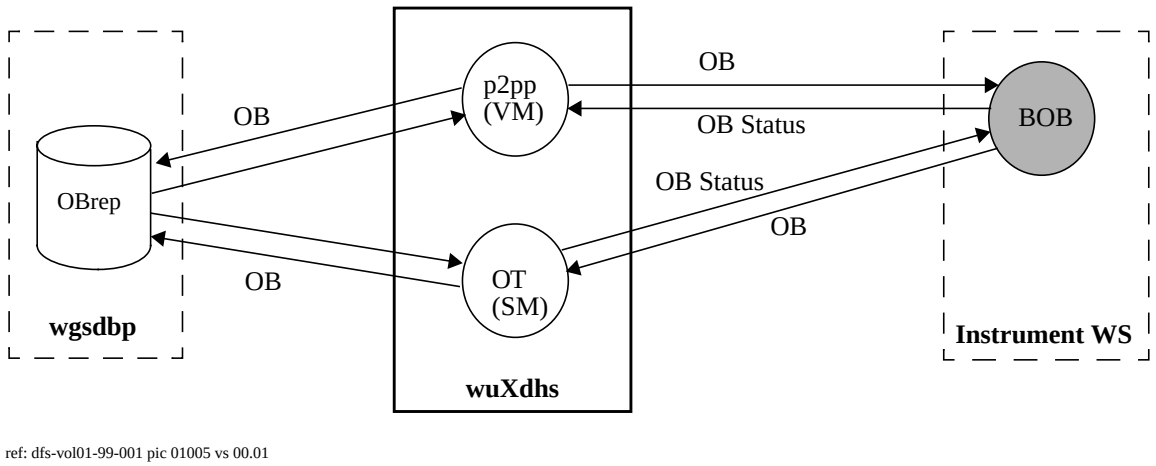
- access the OB repository, via an OB repository browser, to append OBs to a queue or see to which existing queues an OB is already attached to
- order OBs within a queue
- save a queue for later ordering and execution

- add OBs from one or several queues to an "execution sequence" (BOB fetching one or several OBs at a time from this execution sequence). Notice that OT can send several OBs in parallel to BOB (while P2PP send one OB at a time)
- manually update the status of a given OB
- reports generation: ObsBlocks breakdown, Execution Time, OB Verification and instrument configurations reports

A general presentation of OT can be found at http://www.eso.org/~amchavan/papers/spie-2000.ps

- **Interfaces**

Both P2PP and OT interface with BOB (Broker of Observation Blocks), which is the VLT Control System tool which receives Observation Blocks and send s them to the Instrument for execution. Each time an OB is sent to BOB, its status is automatically updated and the new value stored into the OB repository. The same when executing the OB. In fact, an OB status is updated through a "life-cycle" (see [2] for more details).



ref: dfs-vol01-99-001 pic 01005 vs 00.01

Figure 3: OHS interfaces

It is also used in "Engineering Mode" which provides same functionalities as P2PP in "Visitor Mode", except that it does not access the OB repository. "Engineering Mode" is to be used to create Calibration OBs, i.e. to acquire reference data such as domeflats, skyflats, biases, comparison lamps, etc. which do not require the observation of an astronomical target.

OT is used to support Service Mode.

The CCS environment is configured in such way that if two OHS tools (2 P2PP, 2 OT, or 1 P2PP + 1 OT) run in parallel, BOB will communicate by default to the *OHS tool which was launched the first.*

3.3.1.2 OLAS Workstation (wuXdhs)

The Olas Workstation holds all processes/tools which provide the following functionality:

- receive raw frames from the instrument workstations
- transferring raw frames to the subscribers
- ingest data about raw frames into the database (frameIngest)

- **Tools & Processes**

dhs is the server task which receives files from different suppliers:

- from the supplier *vcsolac* running on the Instrument Ws: dhs receives the incoming raw frames to transfer them to the running *subscriber* tasks, i.e. to *frameIngest* on the Olas Ws and to *dhsSubscribe* on the User Ws, the Asto Ws or Pipeline Ws.
- from the supplier *vcsolac* running on the VLT Control Ws: dhs receives operations log files.
- from the supplier *vcsolac* running on the ASM: meteorological and seeing records (on UT1 only)

Actually, dhs polls a specific directory where the suppliers puts the file.

dhs main tasks are also:

- the intermediate storage of incoming files (on disks),
- their delivery to the subscribers (dhsSubscribe on the User, the ASTO and the Pipeline Ws),
- the preparation of summary records for these files and their ingestion into the On-Line archive DB: dhs generates an 'archive filename' for FITS frames mainly based on the MJD-OBS keyword value. It also modify the FITS header to add some keywords (like ORIGFILE for the original filename, or a CHECKSUM value).

Finally, dhs can perform the 'backlog' activity: when a subscriber subscribes to dhs, dhs first sends back a list of files already processed during the last UT night. A user can also request a backlog within a specified date and time range. Upon reception, the subscriber will check this list against the directory where it stores its incoming data, and send back to dhs a list of the missing files.

A new version of dhs allows to handle 2 file systems in order to increase the storage capacity. It tries to fill up the first filesystem, then copies new incoming files to the second filesystem, adding in parallel a link in the first filesystem.

More details about the "OLAS 2 file systems" feature and more generally about dhs are available via the dhs man page (type command: **`man dhs`**, on the Olas Ws).

frameIngest "ingests" incoming raw frames and PAF files. It reads keyword values from the header of the incoming FITS files and updates the *Observation* database. PAF files keywords are stored into the *Ambient* database. Like dhsSubscribe, frameIngest processes at start-up all the left-over files from previous run, and subscribes to the dhs task. By default, it requests data arrived for the current UT night, same as dhsSubscribe.

More details about the frameIngest features are available via the frameIngest man page (type command: man frameIngest, on the Olas Ws).

- **Interfaces**

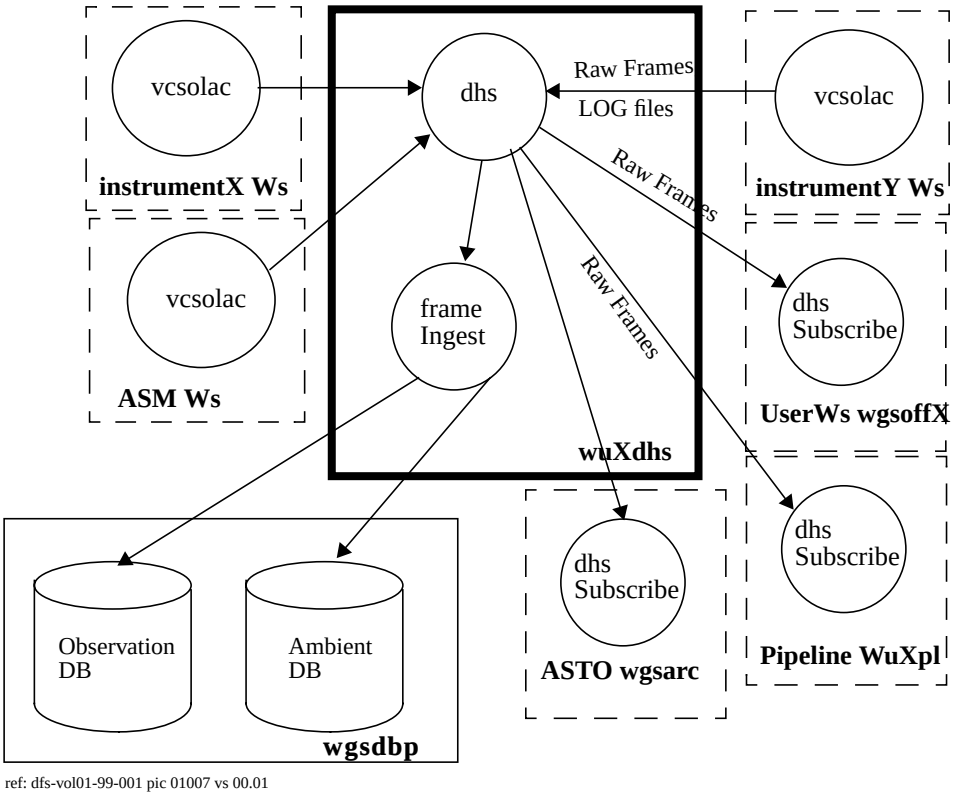


Figure 4: Interfaces of the Olas Ws (wuXdhs)

3.3.2 The User workstations (wgoffX)

3.3.2.1 Tools & processes

dhsSubscribe is an application which sends (via RCP) requests to retrieve specific files.

It can optionally apply a user renaming schema on the received file, or a user-defined command.

Another option, the 'where' clause, allows the user to receive only FITS files containing a specific combination of keywords values.

Several instances of dhsSubscribe can run on the same User Ws, under the same user account, just differentiating them by a -id option.

A subscriber to FITS files will receive FITS tables (.tfits files) as well.

dhsSubscribe is usually not directly called on the User Ws, but rather runs via the *dataSubscriber* application, which is a GUI providing means for configuring, starting and stopping the dhsSubscribers. Two instances of dhsSubscribe usually run on the User Ws: a dhsSubscribe(raw) to subscribe to raw frames, and a dhsSubscribe(reduced) to subscribe to reduced frames.

More details about the dhsSubscribe features are available via the dhsSubscribe man page (type command: man dhsSubscribe, on the User Ws).

gasgano is a GUI which provides following functionality:

- Automatic grouping of files / Organizing and Sorting files

gasgano groups automatically files into virtual directory trees. Users may decide to group the files around the directory structure or around the contents of the files. For the latter, two levels are in all cases being used: files are first organized around the Observing Run to which they belong and then around their parent Observation Block. Users may define a additional third hierarchical level. The grouping is done automatically each time a file is added or moved to the disk.

- Viewing files

  gasgano provides means for viewing those headers, filtering their contents and searching for keywords or keyword values. In addition to this, users may display FITS images.

- Classification of files

  As mentioned before, the archive file names used for ESO data products are unique but user-unfriendly. They in particular do not give any hint to the user about its type and category: bias, dark, etc. Gasgano assigns automatically a classification tag to all FITS files by applying a set of keyword-based logical rules. Those rules are instrument dependent and can be edited/modified by the users. A set of standard rules (the ones used at Paranal and Garching for running pipelines) for all existing VLT instruments is provided as a part of the gasgano package.

- Automatic association of raw files and pipeline products

  Pipeline products are grouped together with the raw and master calibration frames which have been used to generate them, making their relationship intuitively clear.

- Browsing contents of directories

  gasgano provides means for browsing through the virtual directories by filtering their contents to those files whose headers fulfil logical keyword-based expressions. The filtering option is very useful when dealing with a large data set. For instance, it is easy to filter out all calibration frames from a dataset.

- Reporting

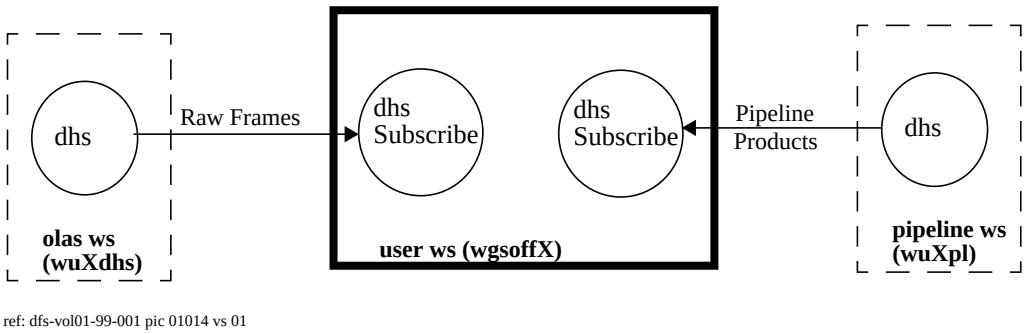
  Gasgano may be used to print reports of selected files or store them as ASCII files on disk. The reports can be configured to the user's need.

- Front-end application to UNIX tools

  gasgano may be used as a front-end application to UNIX tools, e.g. to instrument pipelines. Users may select files and send them as input to an external executable or to a gasgano pre-defined command. The following pre-defined gasgano commands are implemented as button or menu item: report, tar, copy and move.

More details about are available in the Gasgano User's Manual ([11]).

3.3.2.2 Interfaces



ref: dfs-vol01-99-001 pic 01014 vs 01

Figure 5: Interfaces of the User workstations (wgsoffX)

3.3.3 The Pipeline Workstation (wuXpl)

There is one pipeline workstation per UT. This means that each of those machines support at most instrument pipelines, e.g. wu1pl supports the ISAAC and the FORS1 pipelines. The on-line pipeline runs in automatic mode and does not require/authorize any human intervention. Reduction Blocks are created by the Data Organizer and automatically executed by the Reduction Block Scheduler. As a result of the execution of the reduction Blocks, pipeline products are created and QC1 (Quality Control level 1) parameters are generated and stored in the QC1 log files. One log file per instrument is created every day.

3.3.3.1 Tools & Processes

dhsSubscribe subscribes to raw data from wuXdhs. dhsSubscribe is started at boot-time and collects all data created by the current instrument workstation. After processing each frame, it creates symbolic links into a Data Organizer directory which is being polled by that tool.

CalibDB: a local calibration database is installed on each pipeline workstation. This database contain information for the two instruments supported by the workstation. It consists of:

- UNIX directory structure containing classification and reduction rules as well as master calibration files. The structure of the classification and reduction rules is defined in detail in "Data Flow System Specifications for Pipeline and Quality Control"
- msql database which reflects the contents of the directories described above
- a set of commands to update the contents of the calibration database

The **Data Organizer (DO)** is the front-end part of the on-line pipeline. It runs as daemon. It must be initialized by specifying one or more instruments. All the information needed for each of the given instruments is loaded in memory, i.e. master calibration frames are loaded only at starting time.

The DO reacts on the arrival on new raw frames delivered by the dhsSubscribe process described above. It classifies first the incoming FITS files into categories. If applicable for that category, it applies a predefined reduction rule in order to identify the reduction recipe applicable to the frame as well as the master calibration data necessary for executing the recipe. This information is stored in a ReductionBlock that is then delivered to the Reduction Block Scheduler RBS for execution. The DO also checks whether the frame is the last one of the template. In that case and if there is a reduction

rule applicable for this kind of template, it finds out which frames have been created as a result of its execution and creates the associated reduction block. The detection of "end-of-template" is done on the basis of keyword values. It sometimes fails because of missing information. See trouble shooting guide.

The **Reduction Block Scheduler (RBS)** schedules and submits Reduction Blocks to the Data reduction System (MIDAS) for execution.

3.3.3.2 Interfaces

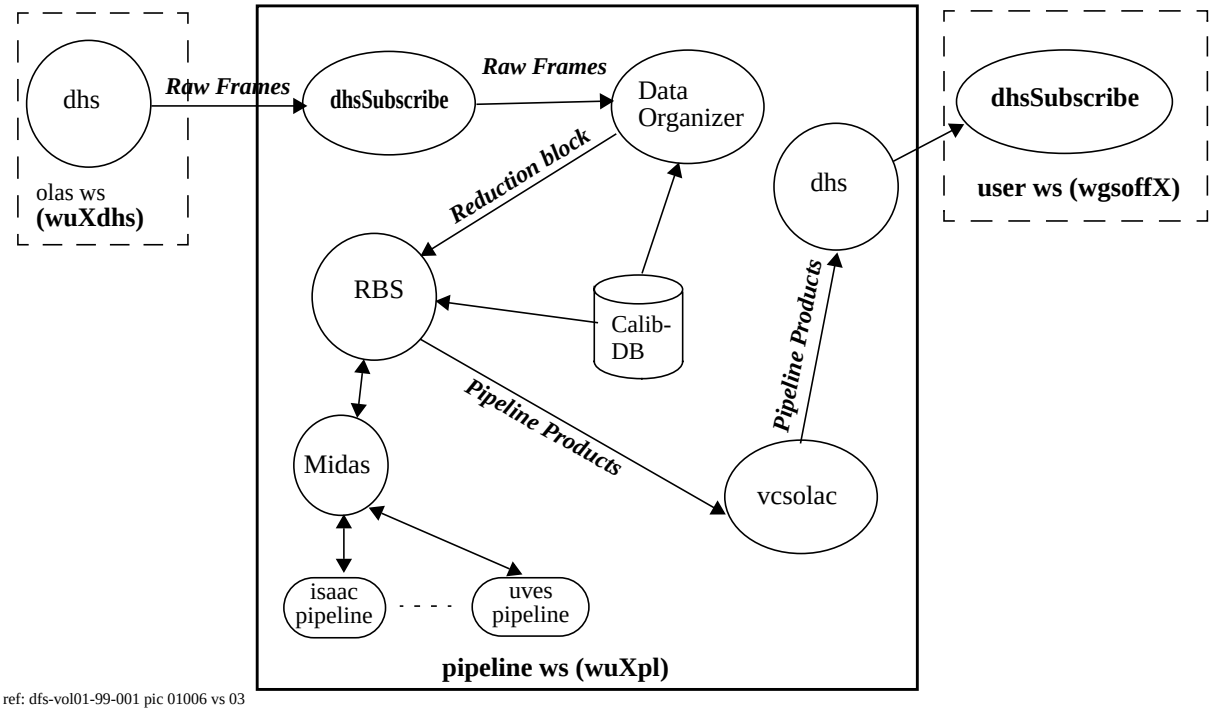


Figure 6: Interfaces for the pipeline workstations wuXpl

3.3.4 The Asto workstation (wgsarc)

There is one ASTO workstation shared by all four UTs. This computer is used for

- archiving the data on long-term storage (DVD). Two copies are being made, one for Garching and one for Paranal).
- preparing CD/DVD for visiting astronomers who have been observing at the VLT.

3.3.4.1 Tools and Processes

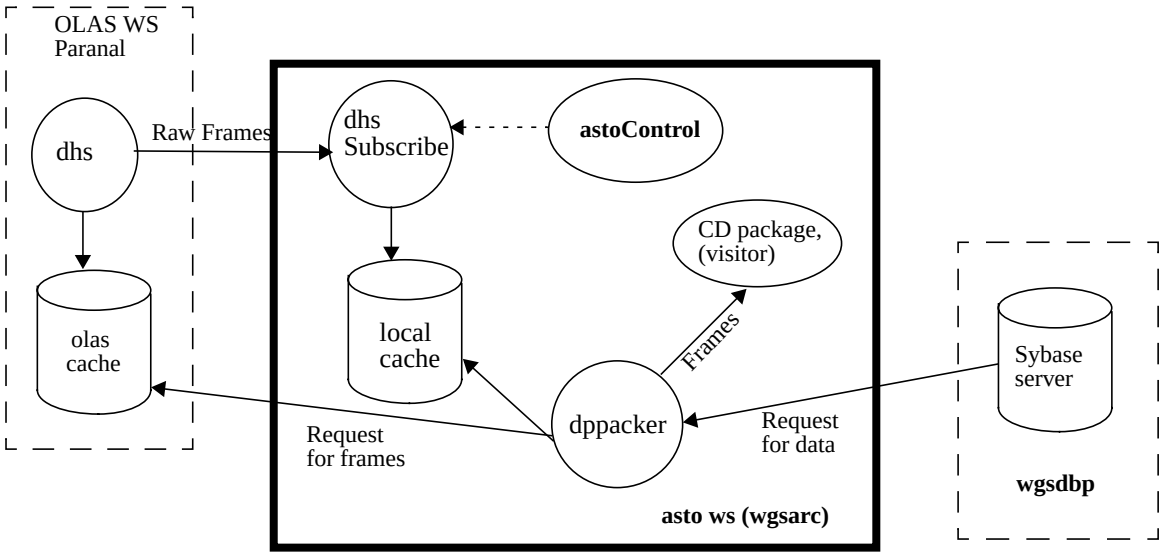
astoControl is a GUI, a front-end interface to asto. astoControl combines all ASTO operations and provides the following functionality:

- retrieve files from wuXdhs through dhsSubscribe
- create media volume

- make media

dppacker is a tool used to set up directory structure containing all of the frames and data for a given visitor mode Observing Run, so that it can be written to CD, tape or other storage media. Given the ProgId of the Observing Run, the tool searches in the database all the frames of category science belonging to the Observing Run and all the calibrations frames which are applicable. The frames, if not present locally are copied from wuXdhs.

3.3.4.2 Interfaces

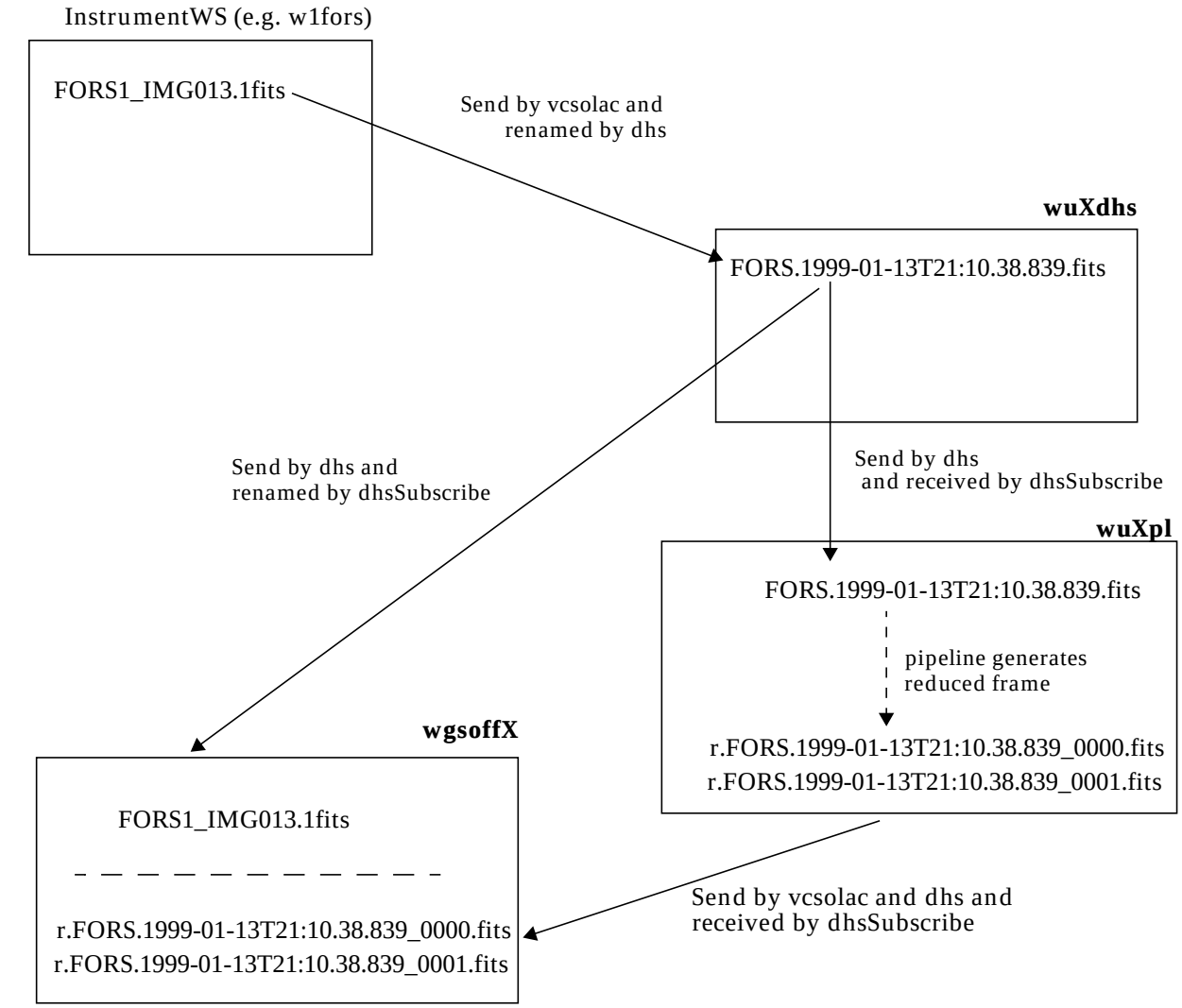


ref: dfs-vol01-99-001 pic 01010 vs 02

Figure 7: Interfaces for the asto ws (wgsarc)

3.4 Frame Renaming Scheme

Raw frames flow through the DFS with a name, the so-called archive file name, which identifies them in a unique way. This name is used on most of the DFS workstations as the UNIX file name on disk. However on the User Ws, which is most visible to the user, the delivered raw frames may get a name chosen after a user-defined scheme, e.g. after a keyword value or a prefix. The following figure shows how a raw frame could be delivered on the user workstation with the same name as on the instrument workstation. Note that pipeline products are not renamed as they go through the system and get delivered to the User Ws.



ref: dfs-vol01-99-001 pic 02001vs 01

Figure 8: The standard renaming scheme

4 Using the Observation Handling tools (wuXdhs)

4.1 Using P2PP

There are mostly two categories of P2PP users at Paranal:

- visiting astronomers, i.e. astronomers who handle *visitor mode OBs*.
- ESO staff astronomers running P2PP in *engineering mode*.

By extension, a *visitor mode OB* is an OB associated to a Visitor Mode Observing Run (and a *service mode OB* is an OB associated to a Service Mode Observing Run).

4.1.1 P2PP for *visitor mode* OBs

Use of P2PP for Visitor Mode OBs is generally done in two steps:

- first, when the visiting astronomer arrives at the mountain: he can run P2PP on a machine which is outside the VLT Control network (i.e. outside the network protected by a firewall), in order to retrieve OBs he may have prepared in advance at his home institute. To do so, he can either re-read a previously created cache directory, or import previously created IMPEX files. This "external" P2PP is configured to access the OB repository server located in Garching, via the AppServer (see previous 3.3.1.1.1 paragraph for details). The astronomer can then modify existing OBs or even create new ones.
  Notice that if the user installs his own cache directory, he will not need to access to the Garching's DB server to create or modify OBs. But, if he simply retrieves IMPEX files, or even creates OBs from scratch, he must access the database server located in Garching in order to get first Observing Runs initially allocated to him by the Phase 1 process. This means that if the DB connection to Garching is not working properly, the visiting astronomer will not be able to use the "external" P2PP: he will have no choice than use P2PP from an OHS Ws (i.e. running on the VLT Control network). In that case, usage of the -noccs option is recommended (see hereafter).
- second, once OBs are ready for execution, they should be transfered from the "external" P2PP to the OHS Ws, either by copying the cache directory, or by exporting IMPEX files.
  The visiting astronomer will then directly control their execution from P2PP running on the OHS Ws (see previous 3.3.1.1. paragraph).

OBs associated to Visiting Mode Observing runs can not be checked-in into the OB repository. Anyway, as soon fetched to BOB, they are automatically saved in the Paranal's OB repository.

To handle visitor mode OBs, P2PP should be started under the UNIX **visitor** account on the **wuXdhs** workstation simply by:

```
p2pp &
```

The tool will then ask to enter a username and password. Each username/password is specific to the visiting astronomer and allows him to access his own Visitor Mode observing runs.

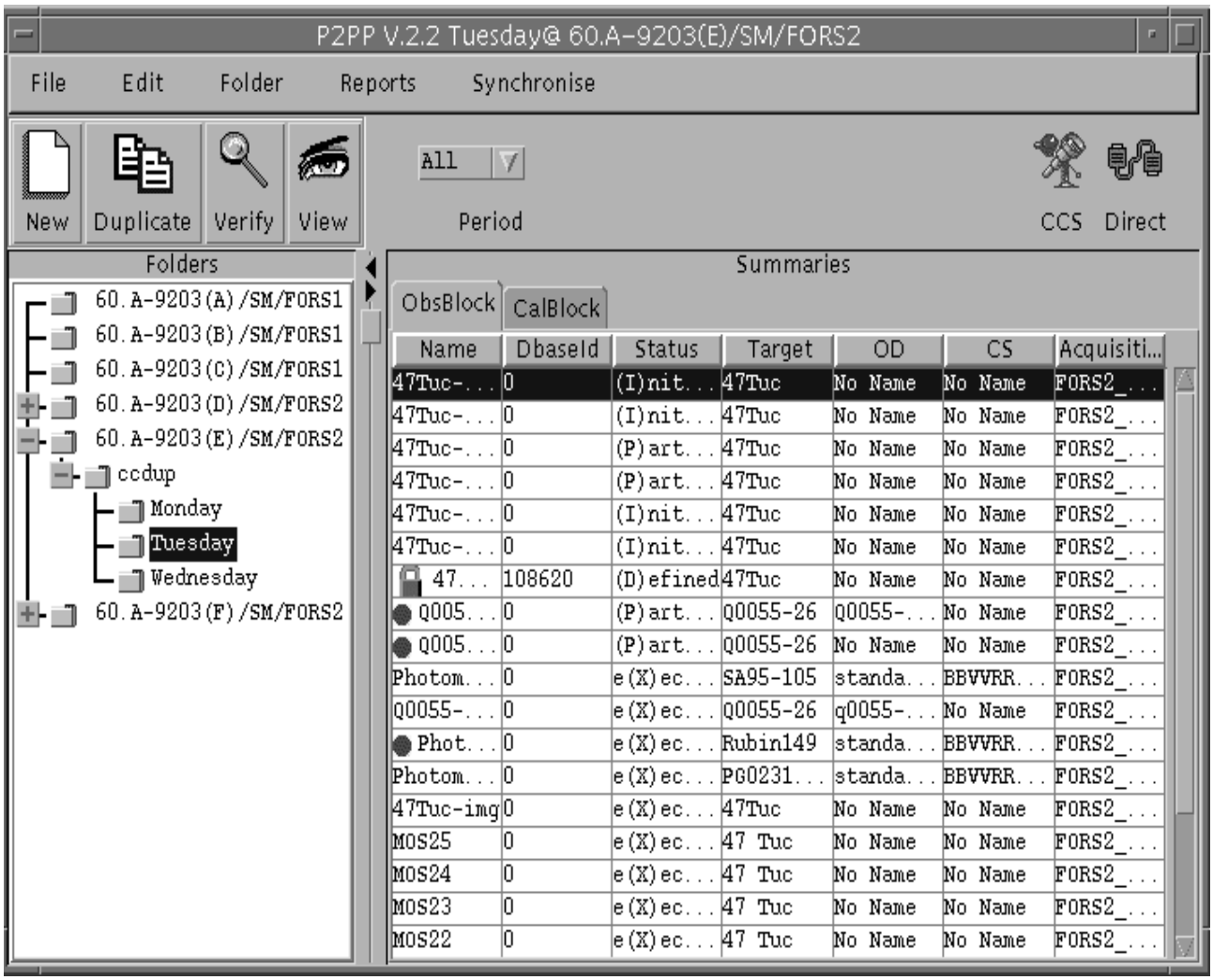
A specific option:

```
p2pp -noccs &
```

can be very useful if it's necessary to create or modify OBs on the OHS Ws, i.e. use P2PP on working OBs, while another P2PP or OT is running in parallel on the same machine and sending "complete" OBs to BOB for execution. The -noccs option will then avoid any interference between concurrent P2PP processes. Indeed, remember that only one P2PP (or one OT) session can talk to BOB at a given time.

Please, refer to the P2PP User's Manual ([2]) for more details about P2PP usage.

Hereafter is the main P2PP window, displaying available observing runs (left side list) for the user, and for the select run, corresponding OBs (right side list).


P2PP V.2.2 Tuesday@ 60.A-9203(E)/SM/FORS2
File
Edit
Folder
Reports
Synchronise
New
Duplicate
Verify
View
All
Period
CCS
Direct
Folders
60.A-9203(A)/SM/FORS1
60.A-9203(B)/SM/FORS1
60.A-9203(C)/SM/FORS1
60.A-9203(D)/SM/FORS2
60.A-9203(E)/SM/FORS2
ccdup
Monday
Tuesday
Wednesday
60.A-9203(F)/SM/FORS2
Summaries
ObsBlock
CalBlock
Name DbaseId Status Target OD CS Acquisiti...
47Tuc-... 0 (I)nit... 47Tuc No Name No Name FORS2_...
47Tuc-... 0 (I)nit... 47Tuc No Name No Name FORS2_...
47Tuc-... 0 (P)art... 47Tuc No Name No Name FORS2_...
47Tuc-... 0 (P)art... 47Tuc No Name No Name FORS2_...
47Tuc-... 0 (I)nit... 47Tuc No Name No Name FORS2_...
47Tuc-... 0 (I)nit... 47Tuc No Name No Name FORS2_...
47... 108620 (D)efined 47Tuc No Name No Name FORS2_...
Q005... 0 (P)art... Q0055-26 Q0055-... No Name FORS2_...
Q005... 0 (P)art... Q0055-26 No Name No Name FORS2_...
Photom... 0 e(X)ec... SA95-105 standa... BBVVRR... FORS2_...
Q0055-... 0 e(X)ec... Q0055-26 q0055-... No Name FORS2_...
Phot... 0 e(X)ec... Rubin149 standa... BBVVRR... FORS2_...
Photom... 0 e(X)ec... PG0231... standa... BBVVRR... FORS2_...
47Tuc-img 0 e(X)ec... 47Tuc No Name No Name FORS2_...
MOS25 0 e(X)ec... 47 Tuc No Name No Name FORS2_...
MOS24 0 e(X)ec... 47 Tuc No Name No Name FORS2_...
MOS23 0 e(X)ec... 47 Tuc No Name No Name FORS2_...
MOS22 0 e(X)ec... 47 Tuc No Name No Name FORS2_...

4.1.2 P2PP in *engineering mode*

Engineering mode is reserved to Paranal's ESO astronomers having to prepare calibration OBs. It also allows to verify the syntax/correctness of TSFs. In this mode, p2pp does not need to access the OB repository. Anyway, it's possible to select an optional feature to check-in related OBs.

Such OBs are sent to BOB as described in previous 3.3.1.1.2 paragraph.

P2PP in engineering mode must be started directly from the UNIX **instmgr** account:

```
p2pp &
```

Be aware that this UNIX account is the only one having write access to the TSFs files (the 'visitor' or 'service' accounts can not do it).

Specific user accounts/passwords are allocated to engineering mode programmes.

Please, refer to the P2PP User's Manual ([2]) for more details about P2PP usage.

4.2 OT for *service* mode OBs

Service mode OBs are prepared by astronomers at their home institute via the P2PP tool. When ready, they need to be stored (checked-in) into the Garching's OB repository. They are then reviewed by the User Support Group.

Finally, these OBs are retrieved by ESO astronomers staff via the OT tool to create queues (see details in previous paragraph 3.3.1.1.). OT also enables to send OBs to BOB for execution.

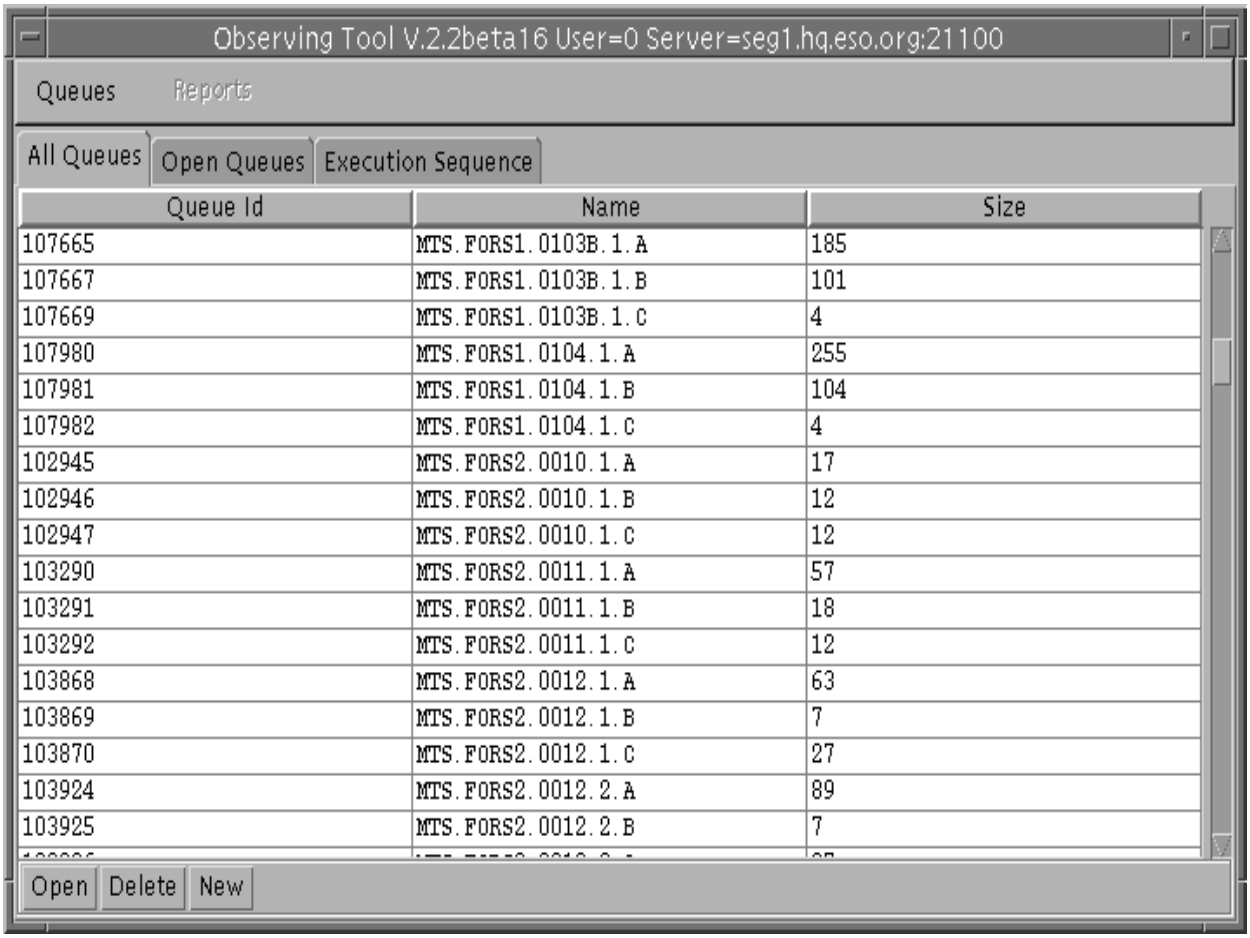
OT must be started under the account **service** on the **wuXdhs** workstation:

```
ot &
```

The tool will then ask to enter specific username and password.

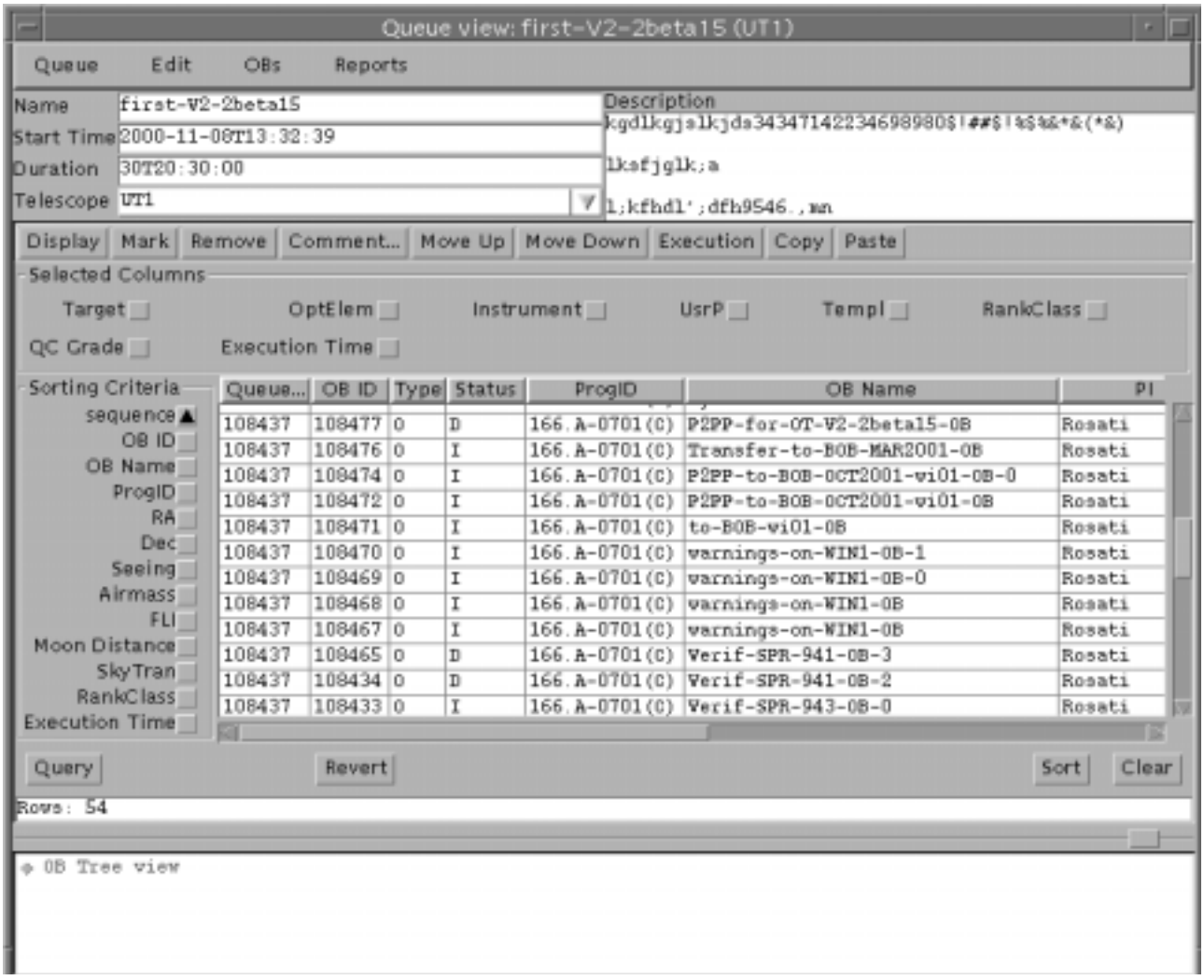
Please, refer to the OT User's Manual ([3]) for more details about OT usage.

Hereafter is an example of the main OT box, listing available queues of OBs:



Queue Id	Name	Size
107665	MTS.FORS1.0103B.1.A	185
107667	MTS.FORS1.0103B.1.B	101
107669	MTS.FORS1.0103B.1.C	4
107980	MTS.FORS1.0104.1.A	255
107981	MTS.FORS1.0104.1.B	104
107982	MTS.FORS1.0104.1.C	4
102945	MTS.FORS2.0010.1.A	17
102946	MTS.FORS2.0010.1.B	12
102947	MTS.FORS2.0010.1.C	12
103290	MTS.FORS2.0011.1.A	57
103291	MTS.FORS2.0011.1.B	18
103292	MTS.FORS2.0011.1.C	12
103868	MTS.FORS2.0012.1.A	63
103869	MTS.FORS2.0012.1.B	7
103870	MTS.FORS2.0012.1.C	27
103924	MTS.FORS2.0012.2.A	89
103925	MTS.FORS2.0012.2.B	7

And an example of the Queue View box, listing OBs associated to a queue:



4.3 Troubleshooting

4.3.1 Problems communicating with BOB

Both OT and P2PP communicates with BOB via the CCS software. If you get an error from BOB like "`OH is not ready to return observation blocks`", it means that OT or P2PP can not properly talk to BOB. Several checks have then to be performed both on the OHS (OH) side, and on the Instrument Workstation. Here are some of them.

The first and easiest one is the following:

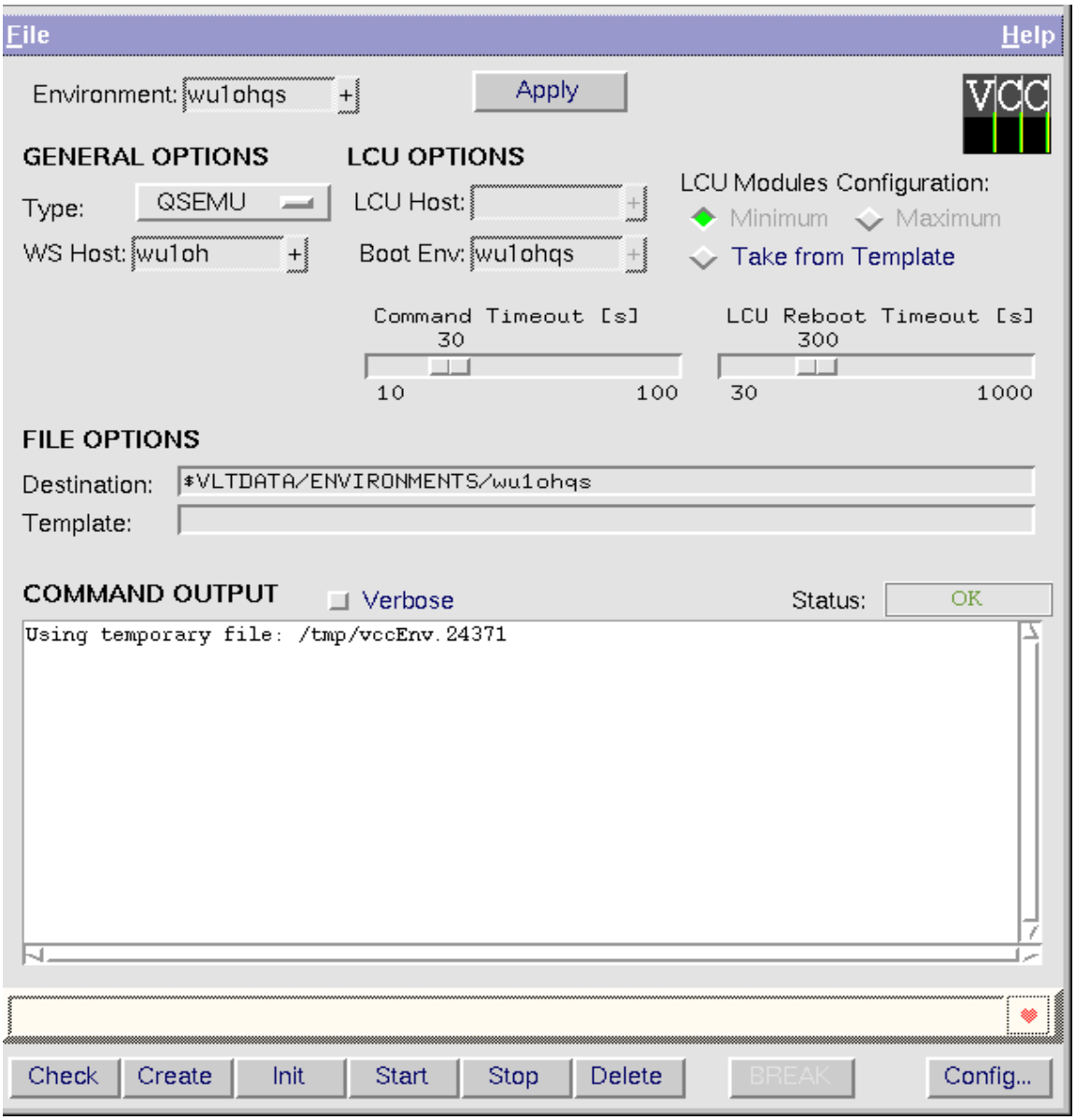
- quit any P2PP or OT session
- check whether CCSLite is running:

ps -ef | grep ccs

should display something like:

```
visitor  6881    1  0  Oct 10  ttyp1    0:00 ccsSHManager -s 20000
visitor  6874    1  0  Oct 10  ttyp1    0:29 ccsScheduler -e wu0ohqs
visitor  6913    1  0  Oct 10  ttyp1    0:00 ccsScan -c 11
```

- If you get no ccsScheduler process, or several ones, it means that the CCS environment has to be re-set:
  - run the vccEnv tool, which is a GUI allowing to control CCS execution (simply type: **vccEnv &**) as **visitor** or **service** account:.



- then via the vccEnv main box, select the environment corresponding to P2PP or OT on the OHS Ws (such environment is usually simply called by the machine name, for instance *wu1dhs*).
- and in sequence, click on the STOP button, then the DELETE one, then CREATE, INIT, and finally on the START button. This should properly stop and re-start the CCS Lite.

- try again to fetch an OB from BOB.

If you still get same error message as above, run the vccEnv tool on the Instrument Workstation on the corresponding CCS environment, and apply same sequence as above.

If the transfer of OBs between P2PP/OT and BOB via CCS still fails, the solution is to save the OB to be executed into an OBD file, and transfer it (via FTP for instance) to the Instrument Ws where BOB runs: a specific BOB menu item allows then to load OBs from a file.

4.3.2 Problems communicating with the OB repository

If P2PP or OT can not communicate with the DB, you usually get an error message from the OHS tool containing the string "`Connection refused`", when starting up the tool.

If the DB connection goes down *while* P2PP or OT is already running, you will get a error message containing the string "`com.sybase.jdbc.SybSQLException`".

In such situation, you have to contact your Database administrator.

4.3.3 BOB complains about a missing template

If during a fetch of OB, BOB complains that a "template xxx is not found", it means that the corresponding TSF file is missing for BOB.

Verify the following directory content: $INS_ROOT/SYSTEM/COMMON/TEMPLATES/TSF

on the Instrument Workstation.

This directory has to be consistent with the place where both P2PP and OT retrieve such files, which is:

- for P2PP, the path given by the INSTRUMENTS.FOLDER keyword in the .p2pp.cf file
- for OT, the given by the INSTRUMENTS.FOLDER keyword in the ~flowmgr/dfs/dataflowJava/config/site.cf

5 Using the User Workstations (wgsoffX)

Raw frames are delivered from wuXdhs to wgsoffX through a subscription mechanism, i.e. by running the dhsSubscribe process. Pipeline products are delivered from wuXpl through the same mechanism by running another dedicated dhsSubscribe.

dataSubscriber is a tool to hide all the complexity of using dhsSubscribe CLI from the user. It allows to:

- Configure the dhsSubscribe processes, including:
  - Defining a renaming scheme for the received frames.
  - Define if and how to copy the keywords in the header of the incoming files. This allows the user to process their frames with data reduction systems which do not understand ESO hierarchical keywords.
- Define the period interval to which the retrieved data should belong, i.e. a "From:" and a "To:" dates and times (see below for formats). If no "To:" date is specified, the subscriber will process all the files created since the "From:" date as well as all newly arriving frames.

During visitor mode observing runs, raw frames and pipeline products belonging to the running Observing Programme as well as any calibration frame may be retrieved. During service mode observations, all frames may be retrieved.

Please note that on each User workstation, four accounts have been set-up, one for each UT: **astro1** to **astro4**. In the remaining part of the chapter these accounts have been generally indicated as **astroY**.

5.1 Subscribing to Raw Frames and Pipeline Products

dataSubscriber is started on **wgsoffX** workstation by account **astroY** in the following way:

`dataSuscriber [-debug {1|2|3} [-log <filename>]]`

where:

`-debug {1|2|3}`: enables the debug mode, where number define the level of verbosity (3 is more verbose than 1). Messages are sent to the standard output.

`-log <filename>`: redirects the debug messages to a log file.

dataSubscriber starts with a subscription window (see picture below) where the user inserts its Observation Program Id (format: PPP.C-NNN(X)) and the observer name. Later, but before starting the subscription processes, it is possible to change this information from the main panel.

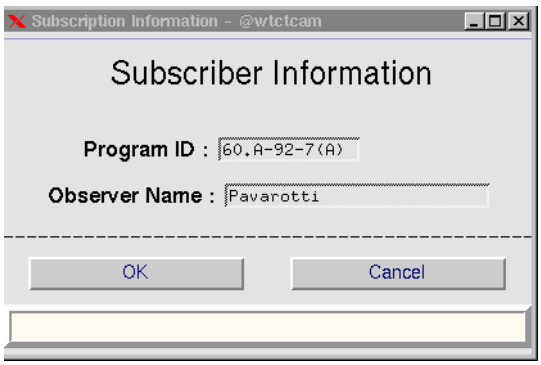


Figure 9: dataSubscriber login window, vs 01

Once the correct information is entered, the main panel appears:

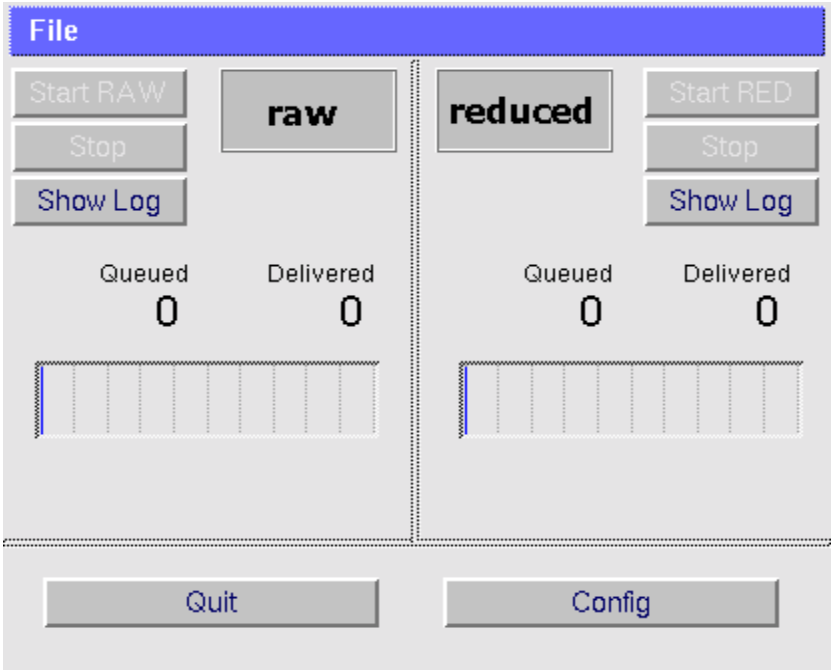


Figure 10: dataSubscriber, main window vs02

By selecting the "File" option, it is possible to:

- Change the Program ID or the Observer name. This option is enabled only if raw and reduced processes are not running.
- View the correspondence between original file names and translated ones.

Before running the subscription processes, the user needs to configure them. The Configuration window appears pressing the "Config" button, and allows the user to insert all required data. Below is a picture of the Configuration window:

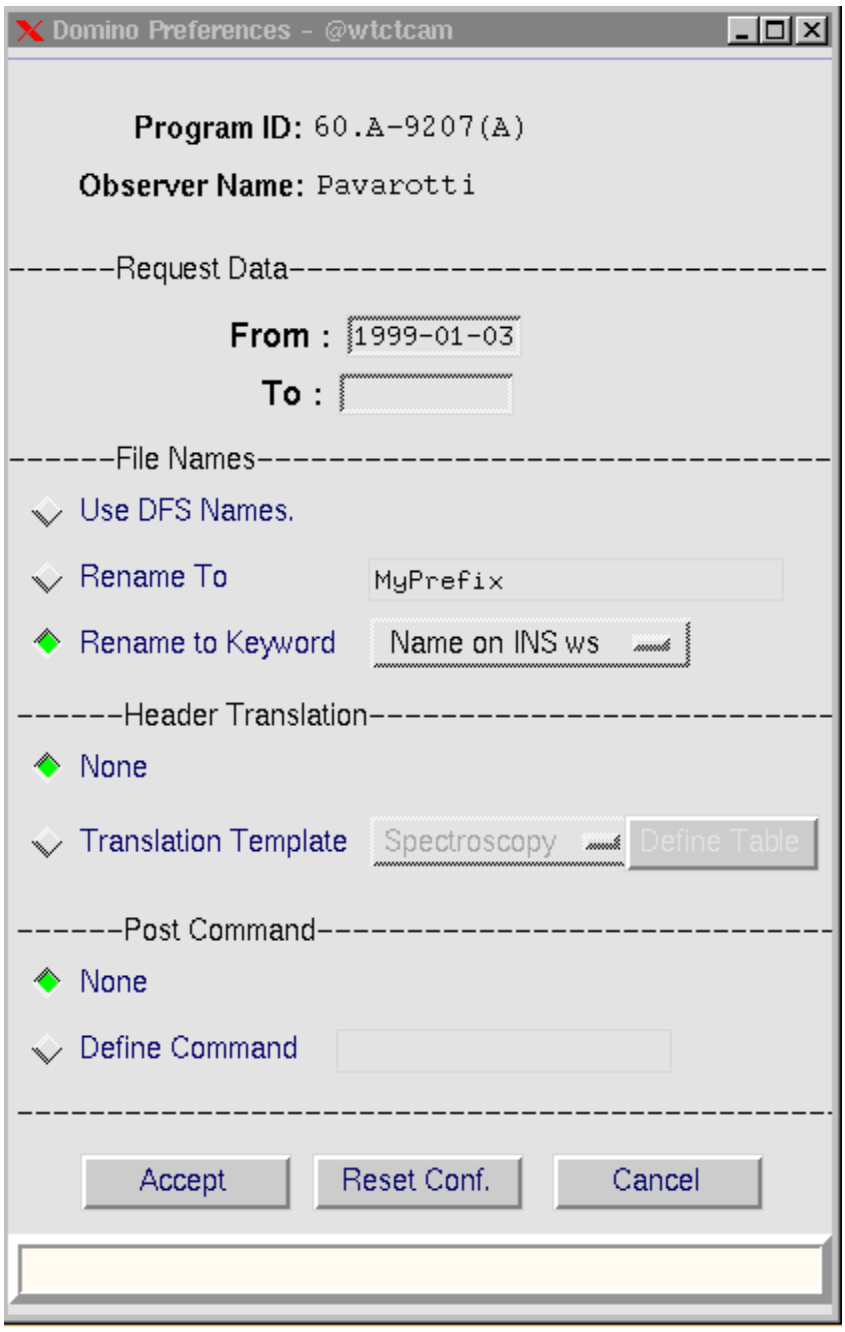


Figure 11: DataSubscriber.PreferenceWindow

The first sub-panel contains the user information, which cannot be changed. The other sub-panels are:

- Request Data: contains the "From:" and "To:" fields, which define a range of dates from which data will be retrieved. The format is: *YYYY-MM-DD*[T*HH*:*MM*:*SS*], i.e. it is possible to specify the exact time up to the second. If time is not specified, the *YYYY-MM-DD* refers to the *observation night*, i.e. to the day of the noon immediately preceeding the observation start time. If the "To:" field is left empty, all files starting from the specified "From:" date will be retrieved.
- File Names: allows the user to:
  - Use DFS names (default): file name are in the standard DFS format.

- Rename To: all the retrieved raw files will have a name starting with the given prefix followed by an incremental number (e.g.MyPrefix_0001.fits, MyPrefix_002.fits).
- Rename to Keyword: the value of the given header keyword is used. Note that the choice is from four predefined keywords.

- Header Translation: it is possible to perform some copies of the header keywords. This option is useful if the retrieved files will be used by programs that do not understand ESO keyword structure. Predefined translation tables are available in the drop-down list, but a custom copy can be defined by pressing the "Define Table" button.
- Post Command: this is the definition of a command to be applied to each file as last step of the retrieval operation. The command is a UNIX command, and can be the call of a data reduction software (e.g., IRAF, MIDAS). Please note that no check is performed on the "safety" of the command itself.

Once configured, it is possible to start the operation separately for raw and reduced data, by pressing the corresponding button in the main window. The progress bar and the "Queued" and "Delivered" fields are updated as the process is working. When all requested file are retrieved, the process stops by itself. Otherwise it is always possible to stop the process by pressing the "Stop" button.

If no "To:" date was specified, the progress bar is greyed out, and the process retrieves any file older then the "From" date. Press the "Stop" button when retrieval is finished.

The retrieved data are put into the following directories on the User Ws:

* /data/raw/*UT-night* for raw FITS files and for PAF files for which a valid UT-night can be determined (for raw files, *UT-night* is a date equal to : the date of MJD-OBS minus 12 hours).

* /data/raw/*current-UT-night* for raw FITS and PAF files for which a valid UT-night can not be determined (*current-UT-night* is the current observation night).

* /data/reduced/*UT-night* for reduced files having a valid MJD-OBS value (where *UT-night* is a date equal to : MJD-OBS date minus 12 hours): in principle, reduced data (.fits and most of the .tfits).

* /data/reduced/*current-UT-night* for other files having no valid MJD-OBS value (where *current-UT-night* is the current observation night): for instance, Reduction Blocks (RBs) or some .tfits files.

Be aware that *UT-night* and *current-UT-night* are usually the same, but not necessarily: for instance, if a raw frame is processed 2 days or more after it has been generated, it will be placed in /data/raw/UT-night, while the corresponding products and RBs will be copied into /data/reduced/current-UT-night.

Also notice that the timestamp of a file (creation or modification timestamp) is not relevant: it does not indicate to which *UT-night* the file belongs to.

For every retrieved file, a soft link is created into the *backlog directory* on the User Ws. The name of this link is the archive filename. The file itself is available in /data/raw or /data/reduced, via its archive filename, or if a file renaming schema is applied, via its user specific name.

5.2 Re-subscribing to Raw Frames or Pipeline Products

This section describes how to resubscribe to raw frames which have already been delivered to wgsoffX in the past. The same 'recipe' is applicable to pipeline products. Let's suppose that the data

obtained on 1999-01-13 have been deleted, i.e. the directory /data/raw/1999-01-13 is empty, or you have chosen a naming scheme that you do not like any more. Should you now start DataSubscriber and start the retrieval process with a "From Date" of 1999-01-13 and "To Date" of 1999-01-13 , no raw frames will be delivered to wgsoffX because dhsSubscribe knows that this set of data has already been delivered. It keeps track of the frames that have been retrieved by creating in a dedicated backlog directory symbolic links to the physical files. The dhsSubscribe process described above will request only the files that are not in the backlog directory.

For instance, let's suppose that the file FORS.1999-01-13T23:10:38.839.fits had been previously processed by dhsSubscribe and renamed to its original name "FORS1_IMG013.1.fits". You should find the following symbolic link in /data/backlog/1999-01-13:

FORS.1999-01-13T23:10:38.839.fits ->
/data/raw/1999-01-13/FORS1_IMG013.1.fits

Therefore in order to re-subscribe to frames which have been already delivered, it is necessary to cleanup the /data/backlog/*creation-date* directory. It is also better to delete the desired files from the /data/raw/*creation-date* directory.

5.3 Managing the currently delivered Raw Frames and/or Pipeline Products

This section describes how the currently delivered data on the User workstation can be checked, regrouped, saved and so on.

All tasks mentioned above will be done with Gasgano, a file organizer. Gasgano is installed for the accounts **astroY** on the **wgsoffX** workstations and can be called in the following way:

```
gasgano &
```

According the preferences, which are set in the last session of Gasgano, the tool will come up as shown in the following picture:

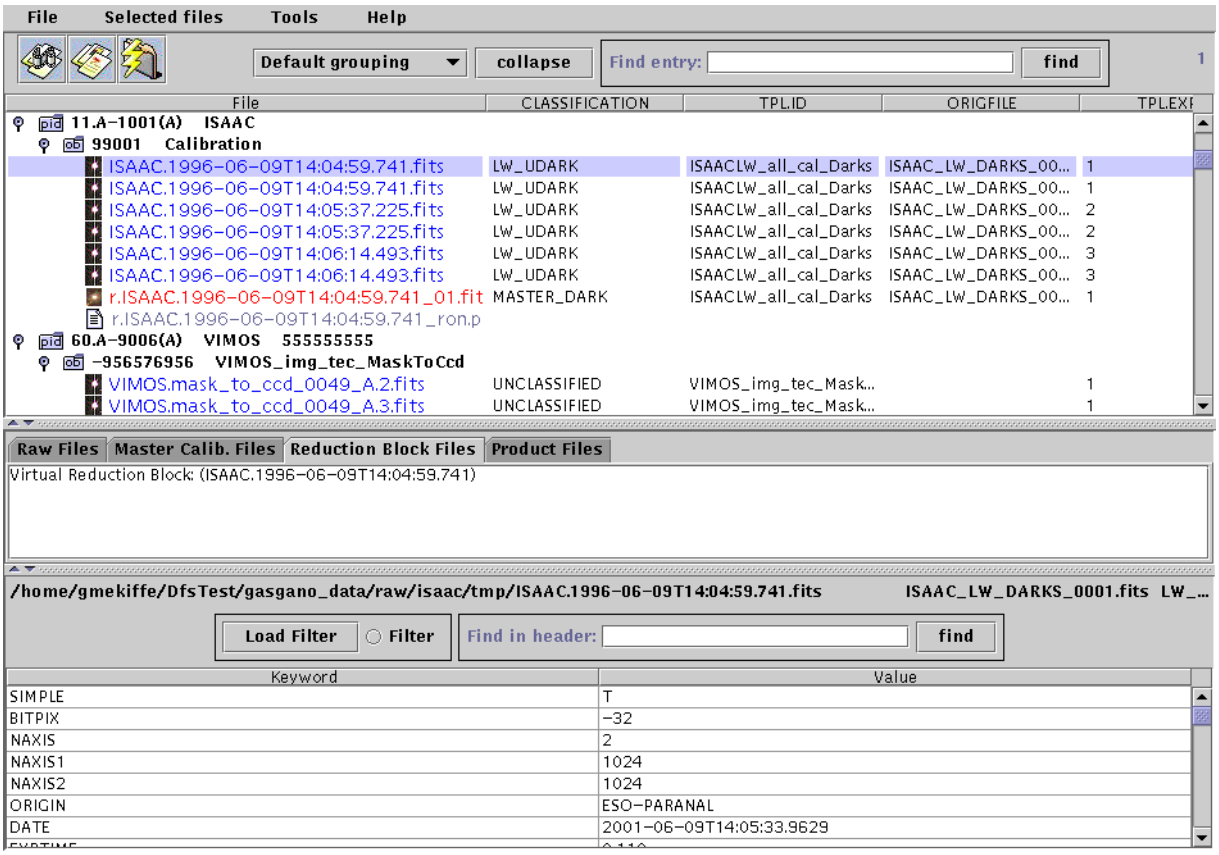


Figure 12: Gasgano, Main window

For more details about Gasgano see the [11].

5.4 Troubleshooting

5.4.1 Raw Frames and Pipeline Products are not delivered any longer

- check the 'dfslog' messages to check any warning ([WARNING] string) or error ([ERROR] string); these messages should explain where the data flow could have been interrupted: at the DHS level on wuXdhs Ws, if DHS correctly transfered the files to the Pipeline Ws, ...

5.4.2 problems to start/stop the RAW and/or REDUCED data subscription

If you get ERROR messages similar to:

- ERROR: DhsSubscribe-wg0off-RAW::wg0off::<OlasWS> is not running. Check the log file
- ERROR: DhsSubscribe-wg0off-RED::wg0off::<PipelineWS> is not running. Check the log file

while <OlasWS> and/or <PipelineWS> are different hostnames than the <OlasWS> and/or <Pipe-

lineWS> names printed in the dataSubscriber GUI:

`RAW files from <OlasWS>`

RED files from <PipelineWS>

it may be due to a corrupted ~/.dataSubscriberConfig file.

In such case, please, stop the 'dataSubscriber' tool and move away the ~/.dataSubscriberConfig

This file will be recreated with the next execution of 'dataSubscriber', and 'dfslog' should not mention such error messages.

5.4.3 Some Pipeline Products seem to be missing

Depending on the current value of an environment variable, all products or some of them only, are delivered to the User Ws. This variable is DFS_PIPE_ALLPRODUCTS and is set on the Pipeline Ws.

If its value is NO, all raw frames will be reduced but only the main products (normally one file, but not always) are delivered. If its value is YES, all types of products will be delivered.

During installation of DFS, the default value is assigned to NO. One can overwritte it manually via a redefinition in the .pecs/misc-all.env file.

5.4.4 dataSubscriber complains not being able to connect to database

When running 'dataSubscriber', if you get a dialog panel with the message"

- "`I cannot get the database password. Call the operator`"

or:

- "`I cannot connect to database. Please check`"

it can be due to:

a) the file ~/.dbrc which does not exist or is not readable, or

b) the file ~/.dbrc exists but it does not contain an entry such as:

`<DBSERVER> asto asto <ENCRYPTED-PASSWORD> ASTOREP`

or is somehow invalid.

<DBSERVER> should give the database server name, e.g.

`ASTOP` at VLT-Paranal, `VLTIP` at VLTI-Paranal, `OLASLS` at La-Silla, `SEGSRV12` in the VCM-Garching or `ESOECF` for DFO-Garching

<ENCRYPTED-PASSWORD> should be the result of executing the command:

```
% stcrypt <PASSWORD>
```

<PASSWORD> for the 'asto' database in <DBSERVER> is known by database operators.

5.4.5 wrong subscription to RAW and REDUCED data

If you get an error message from dataSubscriber similar to

- "`can not get data ...`"

it may be due one or several wrong entries in the configuration file called
~/.dataSubscriberConfig

The entries in this file should match some of your environment variables, e.g.:

DHS_HOST-RAW : $OlasWS
DHS_CONFIG-RAW : $DHS_CONFIG
DHS_HOST-RED : $PipelineWS
DHS_CONFIG-RED : pipeline@$PipelineWS:/data/msg

6 Using the Pipeline Workstations (wuXpl))

6.1 Subscribing to Raw Frames

The raw frame delivery from wuXdhs to wgsoffX is executed through a subscription mechanism, i.e. by running the so-called dhsSubscribe process.

The subscription to raw frames may be started by the account **pipeline** on the wuXpl workstation in the following way:

`dhsSubscribeControl start pipeline`

The subscription to raw frames may be started by the account **pipeline** on the wuXpl workstation in the following way:

`dhsSubscribeControl stop pipeline`

Raw frames, as they are retrieved from wuXdhs get stored in the directory:

/data/raw/*UTnight*

where *UTnight* (YYYY-MM-DD) corresponds to the day of the noon immediately before the corresponding exposure start time. *UTnight* is also usually called *observation night.*

The dhsSubscribe process, after retrieving a frame, creates a symbolic link to the given file in the following directory:

/data/lists/raw_do

For instance, if the file */data/raw/1999-01-13/FORS.1999-01-13T21:10:38.839.fits* has been retrieved from wuXdhs, the following symbolic link is created:

/data/lists/raw_do/FORS.1999-01-13T21:10:38.839.fits.link -> /data/raw/1999-01-13/FORS.1999-01-13T21:10:38.839.fits

6.2 Starting the Data Organizer

The Data Organizer is responsible for classifying any raw frame delivered by the instrument workstation through dhs/dhsSubscribe and for creating reduction blocks if appropriate.

It may be started by the account **pipeline** in the wuXpl workstation on the following way:

`startDO` `[`**`instrument`**`]`
(where *`instrument`* can be a list of values separated by commas but no blank in between)

Note that in the current implementation the Data Organizer is **NOT** started at boot time.

The Data Organizer polls the directory /data/lists/raw_do for symbolic links with the '**.link**' extension. The symbolic links are deleted as soon as the corresponding frames have been processed.

6.3 Starting the Reduction Block Scheduler RBS

The Reduction Block Scheduler executes the reduction blocks created by the Data Organizer. It may be started by the account **pipeline** in the wuXpl workstation in the following way:

`niceRBS` or **`startRBS`**

This command starts the Reduction Block Scheduler in an automatic way, starts the required MIDAS session and creates display windows which will be used later on to display reduced images.

Note that in the current implementation the RBS is **NOT** started at boot time.

6.4 Creating Pipeline Products

The execution of a reduction block may produce one or more products (i.e. FITS frames or FITS binary tables). These products are stored into the directory:

/data/reduced/*creation-date*

where creation-date (YYYY-MM-DD) corresponds to the day of the noon immediately before the exposure start time of the corresponding raw frame.

The names of the pipeline products are built automatically by the Data Organizer. They are based on the name of the corresponding raw frames. For instance, the pipeline reduction of FORS raw frame FORS.1999-01-13T21:10:38.839.fits will result in the creation of n products:

r.FORS.1999-01-13T21:10:38.839.fits_0000.*type*

r.FORS.1999-01-13T21:10:38.839.fits_*nnnn*.*type*

where *nnnn* is determined by the recipe and *type* is a extension describing the type of product, e.g. fits for FITS images, tfits for binary tables, paf for PAF files, log for logfiles.

These pipeline products may be delivered to the User workstation by a combination of OLAS processes (See Section 7.5). The operation team has the option to:

a) deliver all products created by the recipes.

b) deliver only the main product created by the recipes.

This choice has to be made before starting the Reduction Block Scheduler by setting the environment variable DFS_PIPE_ALLPRODUCTS (for all products is 'YES', otherwise the default is 'NO').

Furthermore, the RBS/DRS creates symbolic links to those products in the directory:

/data/lists/reduced_olas

6.5 Supplying Pipeline Products to the User Workstations

The pipeline products created by the pipeline through the execution of reduction block may be delivered to the User Workstation (wgsoffX). This delivery is achieved by running a combination of the three OLAS processes vcsolac, dhs and dhsSubscribe.

vcsolac runs on the pipeline workstation under the account **pipeline**. It may be started in the following way:

```
vcsolacControl start pipeline
```

The delivery of reduced frames may be stopped by the pipeline account in the following way:

```
vcsolacControl stop pipeline
```

Note that in the current implementation, vcsolac is started at boot time.

dhs runs on the pipeline workstation under the account **pipeline**. It may be started in the following way:

```
dhsControl start qc
```

The delivery of reduced frames may be stopped by the pipeline account in the following way:

```
dhsControl stop qc
```

Note that in the current implementation, dhs is started at boot time.

dhsSubscribe is started on the user workstation wgsXoff via the datasubscriber.

6.6 Troubleshooting

6.6.1 The Data Organizer is not receiving any frame

1. Check whether the dhsSubscribe process is running.
2. Check whether the file system /data/raw is full. In this case, it is enough to make some disk space. It is not mandatory to stop dhsSubscribe: this process should recover by itself as soon as disk space is created.
3. Check whether the raw frames you are looking for have been delivered to wuXdhs.

6.6.2 The Data Organizer dies during initialization

You are getting the following error message when starting the Data Organizer:

23:27:19 [INFO] Data Organizer version:DO-1_6

23:27:19 [INFO] Initializing the Data Organizer for isaac

23:27:20 [INFO] Loading the Classification Rules for isaac

** [indexerr] Illegal Index (0) for collection or string with 0
elements

** Processing terminated

1. Check whether the msqld daemon is running (this daemon is normally started at boot time). As **pipline** on wu1pl type:

```
ps -ef | grep msql
```

2. Check whether the environment variables related to calibDB are set correctly.

6.6.3 trouble shooting for the UVES pipeline

Please, refer to B2.16 paragraph of the UVES Pipeline and Quality Control User's Manual [10] to get access to a specific trouble shooting guide for the UVES pipeline.

The following URL
http://www.eso.org/projects/dfs/dfs-shared/web/vlt/vlt-instrument-pipelines.html
gives up to date and interesting information on the UVES pipeline.

7 Using the dfs logging tool dfslog (wuXdhs)

The dfs logging system bases on the sybase database dfslog. This database contains all messages (info, warning and error), which are sent from the several subsystems like vcsolac, dhs, dhsSubscribe, frameIngest, DO, RBs etc.).

To view the data, the "front-end" dfslog should be used. The tool is intended as a light-weight client allowing simple browsing, filtering and display of both recent and archive messages.

More precisely, the DFSlog system is responsible for the logging of the DFS events and messages and provides a GUI for centralized monitoring of DFS operations. It is a means for operators to rapidly receive notification if important events occuring in the DFS. The following subsystems log their information into the dfslog: olas, asto, Data Organizer, Reduction Block Scheduler.

The GUI, called dfslog, allows easy browsing, filtering and reporting of archived messages. In addition, the GUI periodically scans for new data, alerting operators of the arrival of high priority messages (Errors and Warnings).

The dfslog is started a user **archeso** on the **wuXdhs** workstation in the following way:

```
dfslog &
```

If the tool is configured properly, it will come up with a display as shown in Figure 13:

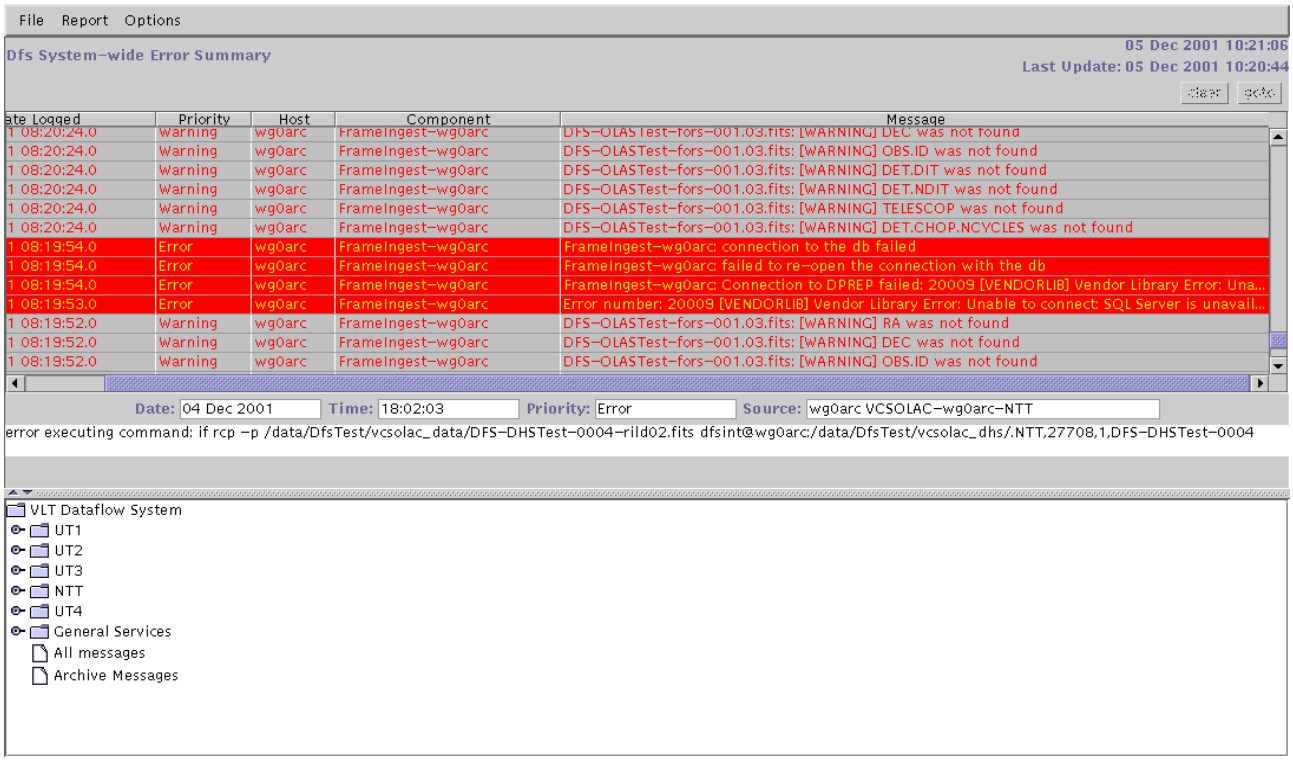

File Report Options
Dfs System-wide Error Summary
05 Dec 2001 10:21:06
Last Update: 05 Dec 2001 10:20:44
Clear
Goto
ate Logged | Priority | Host | Component | Message
1 08:20:24.0 | Warning | wg0arc | FrameIngest-wg0arc | DFS-OLASTest-fors-001.03.fits: [WARNING] DEC was not found
1 08:20:24.0 | Warning | wg0arc | FrameIngest-wg0arc | DFS-OLASTest-fors-001.03.fits: [WARNING] OBS.ID was not found
1 08:20:24.0 | Warning | wg0arc | FrameIngest-wg0arc | DFS-OLASTest-fors-001.03.fits: [WARNING] DET.DIT was not found
1 08:20:24.0 | Warning | wg0arc | FrameIngest-wg0arc | DFS-OLASTest-fors-001.03.fits: [WARNING] DET.NDIT was not found
1 08:20:24.0 | Warning | wg0arc | FrameIngest-wg0arc | DFS-OLASTest-fors-001.03.fits: [WARNING] TELESCOP was not found
1 08:20:24.0 | Warning | wg0arc | FrameIngest-wg0arc | DFS-OLASTest-fors-001.03.fits: [WARNING] DET.CHOP.NCYCLES was not found
1 08:19:54.0 | Error | wg0arc | FrameIngest-wg0arc | FrameIngest-wg0arc: connection to the db failed
1 08:19:54.0 | Error | wg0arc | FrameIngest-wg0arc | FrameIngest-wg0arc: failed to re-open the connection with the db
1 08:19:54.0 | Error | wg0arc | FrameIngest-wg0arc | FrameIngest-wg0arc: Connection to DPREP failed: 20009 [VENDORLIB] Vendor Library Error: Una...
1 08:19:53.0 | Error | wg0arc | FrameIngest-wg0arc | Error number: 20009 [VENDORLIB] Vendor Library Error: Unable to connect: SQL Server is unavail...
1 08:19:52.0 | Warning | wg0arc | FrameIngest-wg0arc | DFS-OLASTest-fors-001.03.fits: [WARNING] RA was not found
1 08:19:52.0 | Warning | wg0arc | FrameIngest-wg0arc | DFS-OLASTest-fors-001.03.fits: [WARNING] DEC was not found
1 08:19:52.0 | Warning | wg0arc | FrameIngest-wg0arc | DFS-OLASTest-fors-001.03.fits: [WARNING] OBS.ID was not found
Date: 04 Dec 2001
Time: 18:02:03
Priority: Error
Source: wg0arc VCSOLAC-wg0arc-NTT
error executing command: if rcp -p /data/DfsTest/vcsolac_data/DFS-DHSTest-0004-rild02.fits dfsint@wg0arc:/data/DfsTest/vcsolac_dhs/.NTT,27708,1,DFS-DHSTest-0004
VLT Dataflow System
UT1
UT2
UT3
NTT
UT4
General Services
All messages
Archive Messages

Figure 13: dfslog, main window with database error

For further details about the dfslog see [12].

7.1 Troubleshooting

7.1.1 dfslog fails to reconnect to the server

The command 'dfslog' is very sensible to the the DatabaseServerURL entry in the ~/.dfslogrc file.

The 'dfslog' reads the ~/.dfslogrc file if it exists, otherwise it is created with default values which can be edited immediately after with an editor panel internal to dfslog.

If the DatabaseServerURL entry is incorrect, dfslog will refuse to start. It will not even open the editor panel, thus the file .dfslogrc has to be edited by hand or removed, in order to force dfslog to create a new one.

With an incorrect DatabaseServerURL entry in the .dfslogrc 'dfslog' will open a message window and exit. The error message may help you to understand the nature of the problem with the DatabaseServerURL entry.

The DatabaseServerURL entry follows this format:

DatabaseServerURL: jdbc:sybase:Tds:<server host>:<port>

a) An error in the <server host> will trigger the following error in'dfslog':

```
Failed to connect to server:
```

java.sql.SQLException:..:UnknownHostException:<server_host>

b) An error in the <port> will produce the following message:

```
Failed to connect to server:
```

java.sql.SQLException:..:Connection refused

c) Any misspelling in the 'jdbc:sybase:Tds' part of the DatabaseServerURL definition will produce the following messages:

```
Failed to connect to server:
```

java.sql.SQLException:..:No suitable driver

or

```
java.sql.SQLException:..:Error loading protocol ...
```

7.1.2 dfslog fails to read in tree graph: treegraph

There is currently a bug in this error message: it is always the same ".treegraph" name whatever the name of your tree graph in the ~/.dfslogrc file is.

Besides that, 'dfslog' only searches the ".treegraph" file in the current directory if your entry (default) in the ~/.dfslogrc file is:

TreeGraphFile=.treegraph

If you want to make 'dfslog' independent of the current directory, open the ~/.dfslogrc file and substitute the ".treegraph" with its absolute path name.

Example:

TreeGraphFile=/diska/archeso/.treegraph

Entries like: "~/.treegraph" or "$HOME/.treegraph are not valid.